



**A University of Sussex DPhil thesis**

Available online via Sussex Research Online:

<http://sro.sussex.ac.uk/>

This thesis is protected by copyright which belongs to the author.

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Please visit Sussex Research Online for more information and further details

UNIVERSITY OF SUSSEX

LIFE SCIENCES

---

**The Stability of Small Atoms and  
Molecules:  
A Quantum Mechanical Three-Body  
Study**

---

*Author:*  
Andrew KING

*Supervisor:*  
Dr. Hazel COX

December 15, 2015



# Abstract

Three-body systems provide the perfect framework for studying the quantum mechanics of both atoms and molecules. These studies can probe the fundamentals of particle interactions that underpin stability, reactivity and structure.

This thesis contains a series of studies into the stability of ground state three-body systems. The focus of this thesis has been the high accuracy computation of three-body systems without recourse to either the Born-Oppenheimer (BO) approximation or approximation of the like-charged particle interaction, which for the case of atoms corresponds to the electron correlation.

Principally the effects of mass and charge on the stability of systems is predicted. The complex nature of coupled electronic interaction is studied to the purpose of pursuing accurate electron correlation that underpins modern computational chemistry.

The energies of three-body systems were calculated very accurately to typically  $\text{mJ mol}^{-1}$  accuracy or better whilst still producing reliable wavefunctions of which all other properties of the system could be calculated accurately. The energies of some of these systems are the lowest to date and all use the latest finite masses as published by CODATA. Computational codes were developed to achieve this accuracy using both numerical and computer algebra methods. These were designed to be efficient, extendable and, importantly, to calculate highly accurate energies, expectation values and wavefunctions.

The masses of any three particles in which there exists at least one bound state below the lowest continuum threshold were identified. The importance of symmetry breaking in a asymmetric system was made clear as the difference in the masses become larger.

A new method was developed to identify the lowest charge of a nucleus that can bind two electrons. This method is more effective than those previously available as it produces a variational upper bound to the true minimum charge in a single calculation. The method was employed to identify the minimum nuclear charge required for binding two electrons in atoms of various nuclear masses. Additionally the electronic structure of such systems was investigated by a judicious partitioning that separates the two electrons into an inner and outer component relative to the nucleus.

The electron correlation was calculated using the Löwdin definition and a highly accurate Hartree-Fock (HF) implementation specifically designed for the task. The effects this electron correlation has on various properties was quantified including the coulomb hole. A second coulomb hole was found which was previously thought to be an artefact but remains even with this highly accurate implementation.

## Acknowledgements

Firstly I'd like to thank my supervisor, Dr Hazel Cox, for all her support and guidance over the years. I would like to thank Professor Tony McCaffery for his encouragement and comments. I would also like to thank Adam Baskerville for the long nights of mathematical pondering. Frank Longford whom performed work in conjunction with that described Chapter 5 and Luke Rhodes for whom performed work in conjunction with Chapter 6.

This work was funded by the University of Sussex and so I extend my gratitude to those who invest into the future.

I would also like to thank those in room 3R511 for a pleasant and entertaining work environment. Of particular note to Sarote Boonseng whom has been a good colleague and friend for the duration of my PhD.

On a personal note I would like to thank those closest in my life, including my family and my little lady, Oraphan Phuangsawai, who have given me the support and motivation I needed to keep going through thick and thin. To these people I am eternally grateful.

# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Overview . . . . .	3
<b>2. Theoretical Background</b>	<b>6</b>
2.1. Atomic Units . . . . .	6
2.2. Eigenvalue Problem . . . . .	7
2.3. Schrödinger Equation . . . . .	8
2.4. Born-Oppenheimer Approximation . . . . .	9
2.5. Electron-Electron Coupling . . . . .	10
2.6. Variational Principle . . . . .	11
2.7. Hartree-Fock Theory . . . . .	12
2.8. The Time Independent Schrödinger Equation for Three-body (3Body) Systems . . . . .	15
2.9. Summary . . . . .	16
<b>3. Computer Programming</b>	<b>17</b>
3.1. Introduction . . . . .	17
3.1.1. Programming Language . . . . .	20
3.1.2. Build Environment . . . . .	26
3.2. C plus plus Programs . . . . .	26
3.2.1. 3Body . . . . .	27
3.2.2. Expectation . . . . .	50
3.2.3. HF2e-Integrals . . . . .	53
3.2.4. Eigen . . . . .	57
3.3. Maple Code . . . . .	58
3.3.1. Maple Recursion Relations . . . . .	58
3.3.2. Other Maple Programs . . . . .	64
3.4. Shell Scripts . . . . .	73
3.4.1. Main Shell Scripts . . . . .	73
3.4.2. Expectation Scripts . . . . .	79
3.5. Conclusions . . . . .	86
<b>4. The Non-Relativistic Ground State Energies of Three-body Systems</b>	<b>87</b>
4.1. Introduction . . . . .	87
4.2. Method . . . . .	89
4.2.1. Perimetric and Triangular Coordinates . . . . .	89
4.2.2. Finite Particle Masses . . . . .	90
4.2.3. General Theory . . . . .	91
4.2.4. Laguerre Polynomials . . . . .	94
4.2.5. Recurrence Relations & Series Solution . . . . .	96
4.2.6. Machine Precision and Numerical Stability . . . . .	99
4.2.7. Asymptotic Behaviour . . . . .	100

4.2.8.	Non-Linear Variational Parameters: The Wavefunction Forms	105
4.2.9.	The Effects of Fermionic Symmetry . . . . .	107
4.2.10.	Pekeris Shell . . . . .	109
4.3.	Implementation . . . . .	109
4.4.	Results and Discussion . . . . .	110
4.4.1.	The Effect of the Non-Linear Variational Parameters . . . . .	110
4.4.2.	The Energy Surfaces of the Non-Linear Parameters . . . . .	121
4.4.3.	Quality of the Wavefunction and Expectation Values . . . . .	123
4.5.	Summary & Conclusions . . . . .	131
<b>5.</b>	<b>The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems</b>	<b>133</b>
5.1.	Introduction . . . . .	133
5.2.	Method . . . . .	136
5.2.1.	Ternary Diagram Explanation . . . . .	137
5.2.2.	Method 0: Energy as a Function of Mass . . . . .	139
5.2.3.	Excess Binding Energy . . . . .	140
5.2.4.	Coordinate System . . . . .	141
5.2.5.	Method 1: Mass Symmetry Breaking Using the Symmetric Ground State Wavefunction . . . . .	141
5.2.6.	Method 2: Variational Principle for the Critical Mass . . . . .	144
5.3.	Implementation . . . . .	146
5.4.	Results & Discussion . . . . .	148
5.4.1.	Method 1: The Calculation of the Stability Domain . . . . .	148
5.4.2.	Method 2: The Variational Stability Domain . . . . .	154
5.4.3.	Comparison of Lower Bound and Variational Methods of the Stability Domain . . . . .	161
5.4.4.	Comparison to Literature: The Existence of $e^+e^-p^+$ . . . . .	161
5.5.	Summary & Conclusions . . . . .	162
<b>6.</b>	<b>The Stability of <math>^1S</math>-States for Coulomb 3-Body Atoms as a Function of Charge</b>	<b>164</b>
6.1.	Introduction . . . . .	164
6.2.	Method . . . . .	167
6.2.1.	Critical Charge . . . . .	168
6.2.2.	Distinguishing the Electrons: Inner and Outer Electrons . . . . .	172
6.3.	Implementation . . . . .	181
6.4.	Results and Discussion . . . . .	186
6.4.1.	Energy as a function of nuclear charge . . . . .	186
6.4.2.	Critical Charge required for binding . . . . .	187
6.4.3.	Bound state properties as a function of nuclear charge . . . . .	191
6.4.4.	Inner and Outer Electrons . . . . .	196
6.5.	Summary & Conclusions . . . . .	201
<b>7.</b>	<b>Electron Correlation in Two Electron Atoms</b>	<b>203</b>
7.1.	Introduction . . . . .	203
7.2.	Method . . . . .	207
7.2.1.	The Hartree-Fock Method . . . . .	207

7.3.	Implementation . . . . .	211
7.3.1.	Gaussian Program . . . . .	211
7.3.2.	Maple Program . . . . .	211
7.4.	Results & Discussion . . . . .	222
7.4.1.	Method Comparison . . . . .	223
7.4.2.	Convergence . . . . .	224
7.4.3.	Non-linear Variational Parameters . . . . .	226
7.4.4.	Hartree-Fock Energies . . . . .	228
7.4.5.	Low Nuclear Charge Hartree-Fock . . . . .	230
7.4.6.	Total Electron Correlation Energies . . . . .	233
7.4.7.	Effect of Correlation on Expectation Values . . . . .	236
7.4.8.	Electron Densities and Electron Holes . . . . .	244
7.4.9.	Hartree-Fock Critical Nuclear Charge . . . . .	247
7.5.	Conclusion . . . . .	248
<b>8.</b>	<b>Summary and Conclusions</b>	<b>251</b>
<b>A.</b>	<b>Numbering Scheme</b>	<b>254</b>
<b>B.</b>	<b>Energy Scaling and Charge Inversion</b>	<b>255</b>
<b>C.</b>	<b>Concavity of the Ground State</b>	<b>256</b>
<b>D.</b>	<b>Arbitrary Nuclear Charge in Gaussian</b>	<b>257</b>
<b>E.</b>	<b>HF 2 Electron Integral Symmetry</b>	<b>258</b>
<b>F.</b>	<b>Licences</b>	<b>259</b>
F.1.	Grand Central Dispatch - Apache License . . . . .	259
F.2.	MPFRC++ - General Public License . . . . .	259
F.3.	MPFR - lesser General Public License . . . . .	261
F.4.	Doublepresso - MIT License . . . . .	262
F.5.	ASMJit - Zlib License . . . . .	262
F.6.	Numerical Recipes Personal Single-User License . . . . .	263
F.7.	Lapack - Modified BSD License . . . . .	264
F.8.	Boost Software License . . . . .	266
F.9.	MacOpt - lesser General Public License . . . . .	267
F.10.	GALib - MIT License . . . . .	268

## List of Acronyms

**2Body** two-body

**2D** 2-dimensional

**3Body** three-body

$a_i$  normalised reciprocal mass

**a.k.a** also known as

<b>AME</b>	Atomic Mass Evaluation
<b>ASM</b>	assembled machine code
<b>bash</b>	Bourne-Again shell
<b>BOBYQA</b>	bound optimisation by quadratic approximation
<b>BO</b>	Born-Oppenheimer
<b>C++</b>	C plus plus
$\delta_{\text{cr}}$	critical delta
<b>DFT</b>	density functional theory
<b>DIIS</b>	Direct Inversion of Iterative Space
$E_{\text{corr}}$	electron correlation
<i>et al.</i>	et alii
$E_{\text{th}}$	threshold energy
<b>FC</b>	fully-correlated
$\hat{F}$	Fock operator
<b>GA</b>	Genetic Algorithm
<b>GUI</b>	graphical user interface
$g$	fractional excess binding energy
<b>HF</b>	Hartree-Fock
<b>Hy-CI</b>	Hylleraas configuration-interaction
<b>IDE</b>	integrated development environment
<b>iff</b>	if and only if
<b>J. Chem. Phys.</b>	Journal of Chemical Physics
<b>JIT</b>	just in time
$L_m(x)$	Laguerre polynomial
<b>MCDF</b>	multi-configurational Dirac Fock
<b>MCHF</b>	multi-configurational Hartree-Fock
$m_{\text{cr}}$	critical mass
<b>MM</b>	molecular mechanics
<b>MR3</b>	multiple relatively robust representations



<b>NLP</b>	non-linear variational parameter
<b>NROpt</b>	Numerical Recipes optimisation
<b>OO</b>	object orientated
<b>pd</b>	particle density
<b>Phys. Rev. A</b>	Physical Review A
<b>QM</b>	quantum mechanics
<b>SCF</b>	Self Consistent Field
<b>sv</b>	save vector
$E^{\text{trial}}$	trial ground state energy
$\psi^{\text{trial}}$	trial wavefunction
$E^{\text{exact}}$	true ground state energy
$\psi^{\text{exact}}$	exact wavefunction
$\psi$	wavefunction
$\oplus$	exclusive or
$Z_{\text{cr}}$	critical nuclear charge

# 1. Introduction

The use of computations in chemistry is of growing significance. Compounds that are too expensive, dangerous or difficult to be feasibly investigated by the experimentalist can be examined at leisure by the computational chemist. However computational calculations need to be as accurate as experiment, and give reliable “real-world”, investigable results. A theoretical investigation should complement the experiment to provide both matching analysis and deeper understanding and ultimately prediction.

Computational chemistry however is not always able to deliver, and is only as good as its underlying mathematics, approximations and computational implementation. In theory, computational chemistry has the tools required to calculate everyday chemical phenomena, however it does not always have the raw computational power to get the job done. To push the limits of feasible calculations approximations are made typically in a proportional manner to the complexity of the problem. Very large systems such as proteins require gross approximations such as molecular mechanics (MM), whereas small molecules use the much more accurate quantum mechanics (QM). Intermediate systems such as metal complexes may use density functional theory (DFT) and even smaller compounds such as aromatic rings, might use wavefunction based mechanics. Within QM in conventional computational chemistry codes such as Gaussian and ADF, the implementations share common approximations. These approximations include the Born-Oppenheimer or fixed nucleus approximation and various approximations on the electron-electron interactions. The Born-Oppenheimer and fixed nucleus approximations assume that the nuclei are fixed and that the electrons move around them in their fixed field. The nuclear and electronic motions are treated separately.<sup>[1]</sup>

Conventional commercial software such as Gaussian also assumes a simplified approximation to the electron-electron interaction, such as that in Hartree-Fock

## 1. Introduction

(HF) method, which treats each electron as a two-body system of nucleus and electron moving in the average field of all other electrons. A great deal of effort is then applied to recover the electron-electron interaction without losing too much of the computational economy bought with this approximation. To go beyond these approximations and observe how systems behave without approximation, requires small systems. Three-body systems are the smallest systems in which the electron correlation of an atom can be calculated and the smallest molecular system in which it is possible to include nuclear motion. Three-body systems are ideal for these accurate computations without approximations. They contain the most important aspects that are often approximated away in conventional computational chemistry codes and are sufficiently simple to calculate fully.

Three-body systems are defined in this work as any system constituted from three particles. The particles can be whole nuclei, such as the helium nucleus or hadrons such as protons ( $p^+$ ) or leptons such as electrons ( $e^-$ ) and its heavier cousin the muon ( $\mu$ ) and their anti-particles. Three particle systems have interesting properties in their own right. For example the three-body system constituted from deuteron, ( $D^+$ ) triton ( $T^+$ ) and muon ( $\mu$ ), is a molecular system but with the much heavier muon particle instead of an electron. It is known to be able to catalyse nuclear fusion at room temperature.<sup>[2–6]</sup> Since the muon is much heavier than an electron the muonic molecular ions ( $\mu DT^+$ ,  $\mu H_2^+$  and so on) have a much shorter bond distance, than for example  $DT^+$ , and this catalyses the fusion between  $D^+$  and  $T^+$ . Use of muon catalysed fusion is restricted however by the energy required to create the muons and the rate at which they are lost to processes such as muon decay and the muons sticking to the He nucleus where it can not catalyse further reactions until “unstuck”.<sup>[3,4,7]</sup> Early estimates suggested that  $\approx 200$  fusion reactions must be catalysed before the muon is lost, to recover the energy required to form the muon. However this value has been shown to be 12 to 15 times too small.<sup>[8–10]</sup> Frolov *et al.*<sup>[10]</sup> calculated the probability of muons sticking to He and the required fusion reactions to break even energetically using three-body calculations. Three-body systems are interesting scientifically as they can be used to explain and probe various aspects of chemistry many of which are still not well understood. One such area where a better understanding would be useful is within the coulomb break-

## 1. Introduction

up problem.<sup>[11–14]</sup> The chemical example of coulomb break-up is electron impact ionisation,<sup>[13]</sup> where an electron impacts a one electron atom and knocks the bound electron off. This problem has at least two asymptotic regions starting with one electron at an asymptotic distance and ending with two electrons at asymptotic distances. The asymptotic behaviour of three-body systems, that describe this dissociation, is not exactly understood and can be quite complex.<sup>[15,16]</sup> The stability of a three-body system to dissociation depends upon many factors which can be investigated within three-body computations.

The work in this thesis explores various aspects of three-body systems with the goal of achieving greater understanding of these interesting systems and aspects of computational chemistry using the non-relativistic time independent Schrödinger equation without the Born-Oppenheimer approximation and with explicit full inclusion of the like-charge interaction. In the case of atoms this is the electron-electron interaction. Performing computations so accurately without approximations has a computational price. Calculations are much slower, however a much greater degree of accuracy can be achieved. The computational cost can be significantly reduced by a judicious use of Laguerre polynomial ( $L_m(x)$ ) wavefunctions ( $\psi$ ) with recursions relations and series solution. Wherever possible this path is taken to minimise computational cost, allowing for larger more accurate wavefunctions to be employed. When the calculations are done in full, without approximations, it is possible to truly understand the underlying assumptions used in other computational codes. The results become more reliable and accurate properties can be investigated. Without the Born-Oppenheimer approximation more complex interactions can be investigated such as the mass effects on stability and the properties of exotic systems such as the Positronium negative ion ( $\text{Ps}^- - e^- e^- e^+$ ) where the Born-Oppenheimer approximation is no longer valid.

### 1.1. Overview

This thesis calculates the energies, wavefunction and expectation values of three-body systems with high accuracy, with energies typically calculated to the  $\text{mJ mol}^{-1}$  or better. In Chapter 4 these properties are calculated and presented. For some of the

## 1. Introduction

more exotic systems, these are the most accurate values of these properties available to date. The computations are benchmarked and compared to available literature and checks on the quality of the wavefunction such as the virial and cusp conditions are performed. This is a vital first step to ensure that accuracy is achieved.

In Chapter 5, the first results chapter, the particle-in-motion design of this methodology permits the exploration of the stability of three-body systems with respect to dissociation to two-body species and a free particle. The masses are shown to affect stability of such a system and that some mass ratios are deemed unstable whilst other ratios are identified as stable to dissociation. A general formula is given to determine if a system has at least one bound state below the lowest continuum threshold of a unit charge three-body system with any given two masses.

In the following Chapter 6 the stability of atoms with varying nuclear charge is studied. A novel method of calculating the minimum nuclear charge to bind two electrons to a nucleus is proposed, developed and implemented. The technique is general and could be expanded to other atoms and molecules. This method has several advantages over the previous methods including its variational nature and its greatly improved computational cost. Additionally the inner and outer electrons are probed using an interesting technique. This is the first time this technique has been applied to a fully correlated wavefunction and it is used to give further insight into how the electrons arrange themselves inside the atom with particular attention to how the electrons behave upon detachment.

Chapter 7 contains one of the most interesting aspects in computational chemistry that is only directly researchable with systems greater than two particles in size, the electron correlation. To achieve this, in addition to our accurate, fully correlated, three-body method, an uncorrelated method that is as accurate as possible is required as a reference. In this chapter the Löwdin definition was used to define electron correlation, meaning that HF is used as the reference method. To ensure these results were as accurate as possible the HF method was implemented using a Laguerre polynomial wavefunction with computational codes specially written for the task of maximising accuracy. The methods were then sufficiently flexible to calculate electron correlation over a wider range of systems including non-integer, low nuclear charge systems, allowing for a more complete study of electron correlation and the effect of

## *1. Introduction*

nuclear charge. Additionally, the effect of electron correlation on other properties such as the electron-nucleus distance and probability densities were studied. The coulomb hole in the singlet ground state of a two electron atom was categorised and the presence of an additional, secondary, coulomb hole identified.

In Chapter 3 the programs written and used throughout this work are discussed, detailing their operation and their design. The choices of methodologies and the impact this has upon this work is outlined. In addition the various external codes that were employed in this work are discussed explaining the benefits they brought to the work.

## 2. Theoretical Background

### 2.1. Atomic Units

To avoid carrying too many constants throughout the calculations and mathematical discussions, a set of units introduced by Hartree called atomic units (a.u.) are used. These units include most constants that appear time and time again when dealing with atoms and molecules and so are often employed in computational chemistry. The units are constituted from various combinations of atomic quantities such as electron charge ( $e$ ) and mass ( $m_e$ ) and Planck's constant ( $h$ ), or more commonly, the reduced Planck's constant ( $\hbar$ ). The following is a list of these units. This is not a definitive list but includes the units used in this work. A more complete list can be found in Bethe and Salpeter.<sup>[17]</sup> For convenience the conversion to SI units is also given. The conversion factors to SI units are from the latest (2010) CODATA<sup>[18]</sup> recommended values for physical constants and are subject to change as measurement techniques improve.

These first 4 constants are set to unity and all other units are defined using these.

**Charge** The charge of the electron is set to unity

$$(1 = e = 1.602\,176\,565(35) \times 10^{-19} \text{ C}^{[18]})$$

**Mass** The mass of the electron is set to unity

$$(1 = m_e = 9.109\,382\,91(40) \times 10^{-31} \text{ kg}^{[18]})$$

**Coulombs Constant** Coulombs constant is set to unity

$$(1 = k_e = \frac{1}{4\pi\epsilon_0} = 8.987\,551\,787 \times 10^9 \text{ N m}^2 \text{ C}^{-2}^{[18]})$$

**Reduced Planck Constant** Reduced Planck constant is set to unity

$$(1 = \hbar = \frac{h}{2\pi} = 1.054\,571\,726(47) \times 10^{-34} \text{ J s}^{[18]})$$

## 2. Theoretical Background

These following units are derived from combinations of the above 4 constants in atomic units.

**Length** The radius of the first Bohr orbit is unity

$$(1 a_0 = 1 \hbar^2 m_e^{-1} e^{-2} = 0.529\,177\,210\,92(17) \times 10^{-10} \text{ m}^{[18]})$$

**Energy** Twice the ionisation energy of an infinite nuclear mass H atom is unity

$$(1 E_h = 1 e^2 a_0^{-1} = 4.359\,744\,34(19) \times 10^{-18} \text{ J} = 2625.49(96) \text{ kJ mol}^{-1}^{[18]}) \text{ This unit is called a Hartree}$$

Using atomic units separates the results from the accuracy of the currently accepted values of physical constants in terms of kg, metres and so on. Of course any conversion from atomic to SI units must accept the SI values. These atomic units are used throughout this thesis unless otherwise stated.

### 2.2. Eigenvalue Problem

The quantum state of a system may be described by a complex function called the wavefunction ( $\psi$ ). Each dynamical property of a system is represented with a linear operator  $\hat{A}$ .<sup>[19]</sup> This operator is associated with the dynamic variable  $a$ . If a particular property is measured experimentally, even under identical conditions, the results would not necessarily be the same, and would have particular possible values such as  $a_0, a_1, \dots, a_n$ . In quantum mechanics these are the eigenvalues of the operator that represent the observable. With a normalised wavefunction an eigenvalue problem is represented with the following equation.<sup>[19]</sup>

$$\hat{A}\psi_n = a_n\psi_n \quad (2.2.1)$$

Since all results of measurement are real numbers this imposes certain conditions on which operators describe physical properties. As a result, all operators that represent physical observables are Hermitian, but not all Hermitian operators represent properties observable experimentally.<sup>[19]</sup>

In a more general case with a wavefunction that is not necessarily normalised the generalised eigenvalue problem is given in Eq. (2.2.2), where  $\hat{S}$  is the overlap, which



## 2. Theoretical Background

defines the inter-relationship between a set of basis vectors of a quantum mechanical system.

$$\hat{A}\psi_n = a_n\hat{S}\psi_n \quad (2.2.2)$$

### 2.3. Schrödinger Equation

Erwin Schrödinger first proposed what is now known as the Schrödinger equation in 1926.<sup>[20,21]</sup> The equation is fundamental to quantum chemistry as it contains the information about the structure and energetics of a system. Although time can be considered, it is almost always used in its time independent non-relativistic form. In this time independent form stationary states are calculated, the properties of which do not change with time. Stationary states are of prime importance in understanding chemical properties. Such properties are independent of time; additionally excitations are the transfer between stationary states.<sup>[22]</sup> This equation in its time independent, non-relativistic form is as follows:

$$\hat{H}\psi = E\psi \quad (2.3.1)$$

The equation is visually simple but is incredibly useful. It contains the probable particle positions in the  $\psi$  and the energy levels are contained in the eigenvalue  $E$ . The operator  $\hat{H}$  is the Hamiltonian operator and this is a combination of the kinetic energy operator  $\hat{T}$  and the potential energy operator  $\hat{V}$ .<sup>[23]</sup>

$$\hat{H} = \hat{T} + \hat{V} \quad (2.3.2a)$$

$$\hat{T} = -\frac{1}{2} \sum_{i=1}^N \frac{1}{m_i} \nabla_i^2 \quad (2.3.2b)$$

$$\hat{V} = \sum_{i=1}^N \sum_{j>i}^N \frac{Z_i Z_j}{|r_{ij}|} \quad (2.3.2c)$$

$|r_{ij}|$  is the distance between particles  $i$  and  $j$ .

$N$  is the number of particles.

$m_i$  is the mass of the  $i^{\text{th}}$  particle.

## 2. Theoretical Background

$\nabla_i^2$  is the laplacian operator of the  $i^{\text{th}}$  particle. In Cartesian coordinates it has the following form:

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \quad (2.3.2d)$$

In quantum chemistry, typically the Schrödinger equation is discussed in terms of electrons and nuclei. The motions of the electrons and the nuclei are separated into two equations. The following form separates the kinetic and potential operators into electrons and nuclei<sup>[23]</sup>, in atomic units  $\frac{1}{m_e}$  is 1:

$$\hat{H} = \underbrace{-\frac{1}{2} \sum_{i=1}^N \frac{1}{m_e} \nabla_i^2}_{\text{The electron's kinetic energy}} - \underbrace{\frac{1}{2} \sum_{a=1}^M \frac{1}{m_a} \nabla_a^2}_{\text{The nuclei's kinetic energy}} + \underbrace{-\sum_{i=1}^N \sum_{a=1}^M \frac{Z_a}{|r_{ia}|}}_{\text{The electron-nucleus potential energy}} + \underbrace{\sum_{i=1}^N \sum_{j>i}^N \frac{1}{|r_{ij}|}}_{\text{The electron-electron potential energy}} + \underbrace{\sum_{a=1}^M \sum_{b>a}^M \frac{Z_a Z_b}{|r_{ab}|}}_{\text{The nucleus-nucleus potential energy}} \quad (2.3.3)$$

The terms in Eq. (2.3.3) are the terms of a Hamiltonian split in a chemist-like way with the electrons and nuclei. The first term in **red** is the kinetic energy of the electrons, whilst the second in **purple** is the kinetic energy of the nuclei. The third term in **blue** is the electron nucleus potential energy this is the same term present in a two-body atom. The third and fourth terms are the electron-electron and nuclear-nuclear potential energy in **green** and **orange** respectively. Perhaps two of the most commonly approximated of these terms are the **purple** nuclei kinetic energy terms in the Fixed Nucleus and the **green** electron-electron potential energy terms in HF and post HF methods.

## 2.4. Born-Oppenheimer Approximation

While the nuclear kinetic energy term in Eq. (2.3.3) explicitly accounts for the motion of the nuclei, this motion is small compared to that of the electrons. For example even a  $\text{H}^+$  nucleus has a mass ratio  $\approx 1836:1$  compared to an electron. The electrons therefore move very far around the nuclei in the time it takes the nuclei to move a short distance. The electrons “see” the heavy slow moving nuclei as almost fixed charges.<sup>[1]</sup> In 1927 Max Born and J. Robert Oppenheimer<sup>[24,25]</sup> treated

## 2. Theoretical Background

the nuclei as stationary with fixed nuclear repulsion between them and the electrons moving through them as a fixed field of nuclear charge. This approximation proved to be effective at reducing computational cost with a small loss in accuracy. The nuclear-nuclear distances are treated parametrically, the nuclear mass is assumed to be infinite resulting in the nuclear kinetic energy being treated as zero. This is the integral part of Born-Oppenheimer (BO) and the Fixed Nucleus approximation, and the nuclei positions are optimised to a minimum on the energy surface using various optimisation techniques. The remaining electronic Hamiltonian becomes as follows:

$$\hat{H} = \underbrace{-\frac{1}{2} \sum_{i=1}^N \frac{1}{m_e} \nabla_i^2}_{\text{The electron's kinetic energy}} + \underbrace{-\sum_{i=1}^N \sum_{a=1}^M \frac{Z_a}{|r_{ia}|}}_{\text{The electron-nucleus potential energy}} + \underbrace{\sum_{i=1}^N \sum_{j>i}^N \frac{1}{|r_{ij}|}}_{\text{The electron-electron potential energy}} + \underbrace{\sum_{a=1}^M \sum_{b>a}^M \frac{Z_a Z_b}{|R_{ab}|}}_{\text{The nucleus-nucleus potential energy (constant)}} \quad (2.4.1)$$

$R_{ab}$  is the fixed nuclear-nuclear distance between nuclei  $a$  and  $b$  it is not a variable and therefore the nucleus-nucleus term can be treated as a constant for a particular molecular structure.

## 2.5. Electron-Electron Coupling

The electron-electron potential energy term in Eq. (2.3.3) (green) is particularly difficult to calculate and is almost always approximated, with varying degrees of success. Conventional computational chemistry codes approximate by treating the electrons as operating in individual spaces only “seeing” the other electrons in an average way. The Hamiltonian is “decoupled” and the electron-electron interaction approximated. This reduces the problem from an  $N$  electron Hamiltonian to one which can be separated to  $N$  one-electron Hamiltonians.

$$\hat{H} = \underbrace{-\frac{1}{2} \sum_{i=1}^N \frac{1}{m_e} \nabla_i^2}_{\text{The electron's kinetic energy}} + \underbrace{-\sum_{i=1}^N \sum_{a=1}^M \frac{Z_a}{|r_{ia}|}}_{\text{The electron-nucleus potential energy}} + \underbrace{\sum_{i=1}^N \mathcal{V}(r_i)}_{\text{Approximated electron-electron potential energy}} + \underbrace{\sum_{a=1}^M \sum_{b>a}^M \frac{Z_a Z_b}{|R_{ab}|}}_{\text{The nucleus-nucleus potential energy (constant)}} \quad (2.5.1)$$

## 2.6. Variational Principle

A variational approach is often a powerful way of solving a problem, whereas other methods such as perturbation theory may fail such as in He where the first term of the perturbation expansion is insufficiently accurate and the higher terms are difficult to calculate.<sup>[26]</sup> In the variational method any trial wavefunction ( $\psi^{\text{trial}}$ ) will give an energy greater than or equal to the true ground state energy ( $E^{\text{exact}}$ ).<sup>[19]</sup> The better the wavefunction  $\psi^{\text{trial}}$  matches that of the exact wavefunction ( $\psi^{\text{exact}}$ ) the more closely the trial ground state energy ( $E^{\text{trial}}$ ) approaches  $E^{\text{exact}}$ .

If the trial wavefunction  $\psi^{\text{trial}}$  exactly matches the true wavefunction  $\psi^{\text{exact}}$  then  $E^{\text{trial}}$  would also match the true  $E^{\text{exact}}$  (Equation (2.6.1a)).

Dirac notation  $\langle \psi | \hat{A} | \psi \rangle$  is commonly used throughout this work and is a convenient shorthand for  $\int \psi \hat{A} \psi d\tau$  where  $d\tau$  is the appropriate volume element for all of space with implied correct integral ranges for all of space. In some cases throughout the thesis, it is further abbreviated to  $\langle \hat{A} \rangle$ .

$$\langle \psi^{\text{trial}} | \hat{H} | \psi^{\text{trial}} \rangle \geq \langle \psi^{\text{exact}} | \hat{H} | \psi^{\text{exact}} \rangle = E^{\text{exact}} \quad (2.6.1a)$$

In the event  $\psi^{\text{trial}}$  does not equate to  $\psi^{\text{exact}}$ , then  $\psi^{\text{trial}}$  may be expanded in terms of a complete set of normalised orthogonal function  $\psi^{\text{trial}} = \psi_0^{\text{trial}}, \psi_1^{\text{trial}}, \dots, \psi_n^{\text{trial}}$  such that:

$$\psi^{\text{trial}} = \sum_n a_n \psi_n^{\text{trial}} \quad (2.6.1b)$$

where

$$\sum_n a_n a_n = 1 \quad (2.6.1c)$$

Substituting Eq. (2.6.1b) into Eq. (2.6.1a) leads to the following:

$$\hat{H} \psi_n^{\text{trial}} = E_n^{\text{trial}} \psi_n^{\text{trial}} \quad (2.6.1d)$$

$$E^{\text{trial}} = \sum_n \sum_{n'} a_n a_{n'} \langle \psi_n^{\text{trial}} | \hat{H} | \psi_{n'}^{\text{trial}} \rangle \quad (2.6.1e)$$

## 2. Theoretical Background

The trial wavefunction is orthogonal in that  $\langle \psi_n^{\text{trial}} | \psi_{n'}^{\text{trial}} \rangle = \delta_{n,n'}$ . As a result the expanded terms of Eq. (2.6.1e) where  $n$  is not equal to  $n'$  vanish and we are left with:

$$E^{\text{trial}} = \sum_n a_n a_n E_n^{\text{trial}} \quad (2.6.1f)$$

The difference between  $E^{\text{trial}}$  and  $E^{\text{exact}}$  is thus expressed by:

$$E^{\text{trial}} - E^{\text{exact}} = \sum_n a_n a_n (E_n^{\text{trial}} - E^{\text{exact}}) \quad (2.6.1g)$$

Since  $a_n$  is positive or zero and  $\sum_n a_n a_n = 1$  and also that  $E_n^{\text{trial}}$  is always greater than or equal to  $E^{\text{exact}}$ <sup>[23]</sup> it follows that Eq. (2.6.1g) is always positive or zero and that the following is true regardless of the wavefunction:

$$\langle \psi^{\text{trial}} | \hat{H} | \psi^{\text{trial}} \rangle \geq E^{\text{exact}} \quad (2.6.2)$$

No matter what wavefunction is used the energy will never be below the true ground state energy. It is the nature of the Hamiltonian employed here that there is always a lowest energy eigenvalue and so the eigenvalues are said to be bounded from below.<sup>[23]</sup> This means that the possible eigenvalues do not extend to  $-\infty$ , and the lowest eigenvalue itself is said to be an upper bound to the true eigenvalue.<sup>[23]</sup>

## 2.7. Hartree-Fock Theory

The HF approximation is often the first approximation within wavefunction mechanics in computational codes such as Gaussian. The HF approximation treats an  $N$ -electron system as  $N$  one-electron systems only interacting with the other electrons in an average way. The Hartree method was introduced to make a simpler but plausible wavefunction from the product of one electron wavefunctions.<sup>[27]</sup>

$$\Psi = \psi(1)\psi(2)\psi(3) \cdots \psi(n) \quad (2.7.1)$$

## 2. Theoretical Background

This is the Hartree product, not the HF process which we will discuss shortly and is solved more efficiently, where  $\Psi$  is the function of all electrons,  $\psi(1)$  is a function of the coordinates of the first electron and  $\psi(2)$  is a function of the coordinates of the second electron and so on. Each of these  $\psi(i)$  are called either atomic or molecular orbitals. The Hartree process<sup>[27]</sup> was to iteratively solve this system by first optimising electron 1 in  $\psi(1)$  in the average field of all other electrons in  $\psi(2), \psi(3), \dots, \psi(n)$  at this stage only  $\psi(1)$  is changed. After which electron 2 is optimised in the field of electrons  $\psi(1), \psi(3), \dots, \psi(n)$  and so with all electrons up to  $n$  which completes a single cycle. After all electrons are optimised the process is started again and iteratively repeated until the electrons converge (in terms of energy or orbital coefficients) and cease to change by some defined amount.<sup>[27]</sup>

The Hartree method has two flaws originating from its lack of treatment of electron “spin”. One of the consequences of spin is that no more than two electrons can occupy the same orbital. The other consequence is that when two electrons are exchanged the wavefunction changes sign, the wavefunction is said to be antisymmetric to exchange. However the Hartree wavefunction is symmetric to exchange as it does not change sign. These are accounted for by the work of Slater and Fock.<sup>[27]</sup>

The Hartree product in Eq. (2.7.1) is a product of spatial wavefunctions. Slater’s wavefunction however, also contains spin functions corresponding to an electron in an up spin state ( $\alpha$ ) or a down spin state ( $\beta$ ) to give two spin orbitals for each spatial orbital. In the restricted HF model there are two electrons in any one spatial orbital. Here we only discuss this restricted model and therefore in the discussion there are  $2n$  electrons in  $n$  orbitals. The Slater wavefunction differs from the Hartree in that it has these spin function but also in that it is not a simple combination of orbitals. Instead the Slater wavefunction uses a determinant.<sup>[27]</sup>

$$\Psi = \frac{1}{\sqrt{(2n)!}} \begin{vmatrix} \psi_1(1)\alpha(1) & \psi_1(1)\beta(1) & \cdots & \psi_n(1)\alpha(1) & \psi_n(1)\beta(1) \\ \psi_1(2)\alpha(2) & \psi_1(2)\beta(2) & \cdots & \psi_n(2)\alpha(2) & \psi_n(2)\beta(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \psi_1(2n)\alpha(2n) & \psi_1(2n)\beta(2n) & \cdots & \psi_n(2n)\alpha(2n) & \psi_n(2n)\beta(2n) \end{vmatrix} \quad (2.7.2)$$

This is called a Slater determinant and enforces the Pauli exclusion principle, if any two electrons had identical quantum numbers then two rows or columns would be

## 2. Theoretical Background

identical and the determinant and the wavefunction would vanish. Switching two electrons corresponds to switching of two rows and this handles the sign changes.

The HF energy is calculated with the following equation for  $n$  electron pairs and  $M$  nuclei. In the following discussion  $\tau_1$  is the appropriate coordinates and volume for the one electron problem, and  $\tau_2$  is appropriate coordinates and volume for the two electron problem.

$$E = 2 \sum_{i=1}^n \int \psi_i^*(1) \hat{H}^{\text{core}}(1) \psi_i(1) + \sum_{i=1}^n \sum_{j=1}^n \psi_i^*(1) \left( 2\hat{J}_j(1) - \hat{K}_j(1) \right) \psi_i(1) d\tau_1 \quad (2.7.3a)$$

$$\hat{H}^{\text{core}}(1) = \left( -\frac{1}{2} \nabla_1^2 - \sum_{a=1}^M \frac{Z_a}{r_{a1}} \right) \quad (2.7.3b)$$

$$\hat{J}_j(1) = \int \psi_j^*(2) \left( \frac{1}{r_{12}} \right) \psi_j(2) d\tau_2 \quad (2.7.3c)$$

$$\hat{K}_j(1) \psi_i(1) = \int \psi_j(1) \psi_j^*(2) \left( \frac{1}{r_{12}} \right) \psi_i(2) d\tau_2 \quad (2.7.3d)$$

$\hat{J}$  is the coulomb operator and represents the electrostatic repulsion between the charge clouds of electrons.  $\hat{K}$  is referred to as the exchange operator. Although no simple interpretation exists for  $\hat{K}$  like there is for  $\hat{J}$ ,  $\hat{K}$  can be thought of as a correction to  $\hat{J}$  arising from two electrons of the same spin avoiding each other more and effectively reducing the electrostatic repulsion.<sup>[27]</sup>

Equation (2.7.3a) is not quite an eigenvalue equation but rather a sum of eigenvalue equations.<sup>[27]</sup> To solve the problem as an eigenvalue equation the Fock operator is used:

$$\hat{F} = \hat{H}^{\text{core}}(1) + \sum_{j=1}^n \left( 2\hat{J}_j(1) - \hat{K}_j(1) \right) \quad (2.7.4)$$

The Fock operator ( $\hat{F}$ ) can be used to solve HF equations as an eigenvalue problem.  $\hat{F}$  operates on a total of  $n$  spatial orbitals each containing 2 electrons for a total of

## 2. Theoretical Background

$2n$  electrons. Each Fock operator calculates the energy of each spatial orbital<sup>[27]</sup>:

$$\begin{aligned}\hat{F}\psi_1(1) &= \epsilon_1\psi_1(1) \\ \hat{F}\psi_2(1) &= \epsilon_2\psi_2(1) \\ &\vdots \\ \hat{F}\psi_n(1) &= \epsilon_n\psi_n(1)\end{aligned}\tag{2.7.5}$$

Equation (2.7.5) can be written in matrix form which is diagonalisable.

$$\hat{F} \begin{pmatrix} \psi_1(1) \\ \psi_2(1) \\ \vdots \\ \psi_n(1) \end{pmatrix} = \begin{pmatrix} \epsilon_1 & 0 & \cdots & 0 \\ 0 & \epsilon_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \epsilon_n \end{pmatrix} \begin{pmatrix} \psi_1(1) \\ \psi_2(1) \\ \vdots \\ \psi_n(1) \end{pmatrix}\tag{2.7.6}$$

The  $\hat{F}$  operator does not form a true eigenvalue equation because the operator depends upon the wavefunction and is therefore what is referred to as a pseudo eigenvalue equation.<sup>[27]</sup> To solve this problem the equations are solved iteratively from some guessed wavefunction until the wavefunction ceases to change by some reasonable amount. The  $\epsilon_i$  values are the energy levels of the system and not the total electronic energy. The total electronic energy is  $E = \sum_i^n \left( \epsilon_i + \left\langle \psi_i(1) \hat{H}^{\text{core}}(1) \psi_i(1) \right\rangle \right)$ . Further details of this method are given in Chapter 7.

## 2.8. The Time Independent Schrödinger Equation for 3Body Systems

In this work the Schrödinger equation for a three-body system with particle masses  $m_i$  and charges of  $Z_i$  is:

$$\hat{H}\psi = E\psi\tag{2.8.1a}$$

$$\hat{H} = -\frac{1}{2m_1}\nabla_1^2 - \frac{1}{2m_2}\nabla_2^2 - \frac{1}{m_3}\nabla_3^2 + \frac{Z_1Z_3}{r_1} + \frac{Z_2Z_3}{r_2} + \frac{Z_1Z_2}{r_3}\tag{2.8.1b}$$

The centre of mass motion is separated off to give the interaction Hamiltonian.<sup>[17,28]</sup>

We take the origin to be at particle 3 which is the particle with the opposite sign



## 2. Theoretical Background

charge to the other two.

$$\hat{H} = -\frac{1}{2\mu_1}\nabla_1^2 - \frac{1}{2\mu_2}\nabla_2^2 - \frac{1}{m_3}\nabla_1 \cdot \nabla_2 + \frac{Z_1 Z_3}{r_1} + \frac{Z_2 Z_3}{r_2} + \frac{Z_1 Z_2}{r_3} \quad (2.8.2a)$$

where

$$\mu_1 = \frac{m_1 m_3}{m_1 + m_3}, \quad \mu_2 = \frac{m_2 m_3}{m_2 + m_3} \quad (2.8.2b)$$

$$(2.8.2c)$$

This is the Hamiltonian solved in this work. Further details on its separation and solution are given in Chapter 4. This Hamiltonian explicitly includes all particles masses and interactions, treating the Hamiltonian fully in the time independent non-relativistic regime.

## 2.9. Summary

The wavefunction of a system can describe all dynamic properties of a system. For the simplest of systems such as one electron non-relativistic atoms with infinite nuclear mass exact solutions are known. For more complex systems, as in more than one electron, an exact solution is unknown. With the use of approximations however a practical and highly informative field of computational chemistry has developed for these many electron systems. However these approximations need not be made if the system is small enough for example an atom with two electrons can be solved to a high degree of accuracy using the full time independent Schrödinger equation (Eqs. (2.3.1) and (2.3.2)) by treating the mass of the nuclei as finite and incorporating explicitly the inter electron distances.

## 3. Computer Programming

### 3.1. Introduction

Throughout this thesis various pieces of code were written to calculate all aspects of the three-body systems that were investigated, including the recursion relations eigenvalue problem and the expectation values. This chapter discusses the programs and their design, first describing the C plus plus (C++) programs then the Maple programs and bash programs. The main workhorse was a C++ program named **3Body**. The **3Body** program would create the matrices, diagonalise the eigenproblem, optimise the non-linear variational parameters (NLPs) and calculate the energy and the wavefunctions which would then be used elsewhere to calculate other desired properties in other programs. This chapter provides the details of the programs written and used.

During development the design philosophy was to make good use of the available resources and not to “reinvent the wheel”. That is to say that external pieces of codes such as numerical recipes,<sup>[29,30]</sup> Boost<sup>[31]</sup> and various other libraries were used in the creation of this program. Doing so provided many benefits, these external libraries are often faster and they have been optimised for efficiency and additionally they are better maintained such that any bugs are tested for and reported by a much wider audience and in general reduced the development time. This chapter will detail where these pieces of code are used why they were used and how they were brought together to create the various programs in this work.

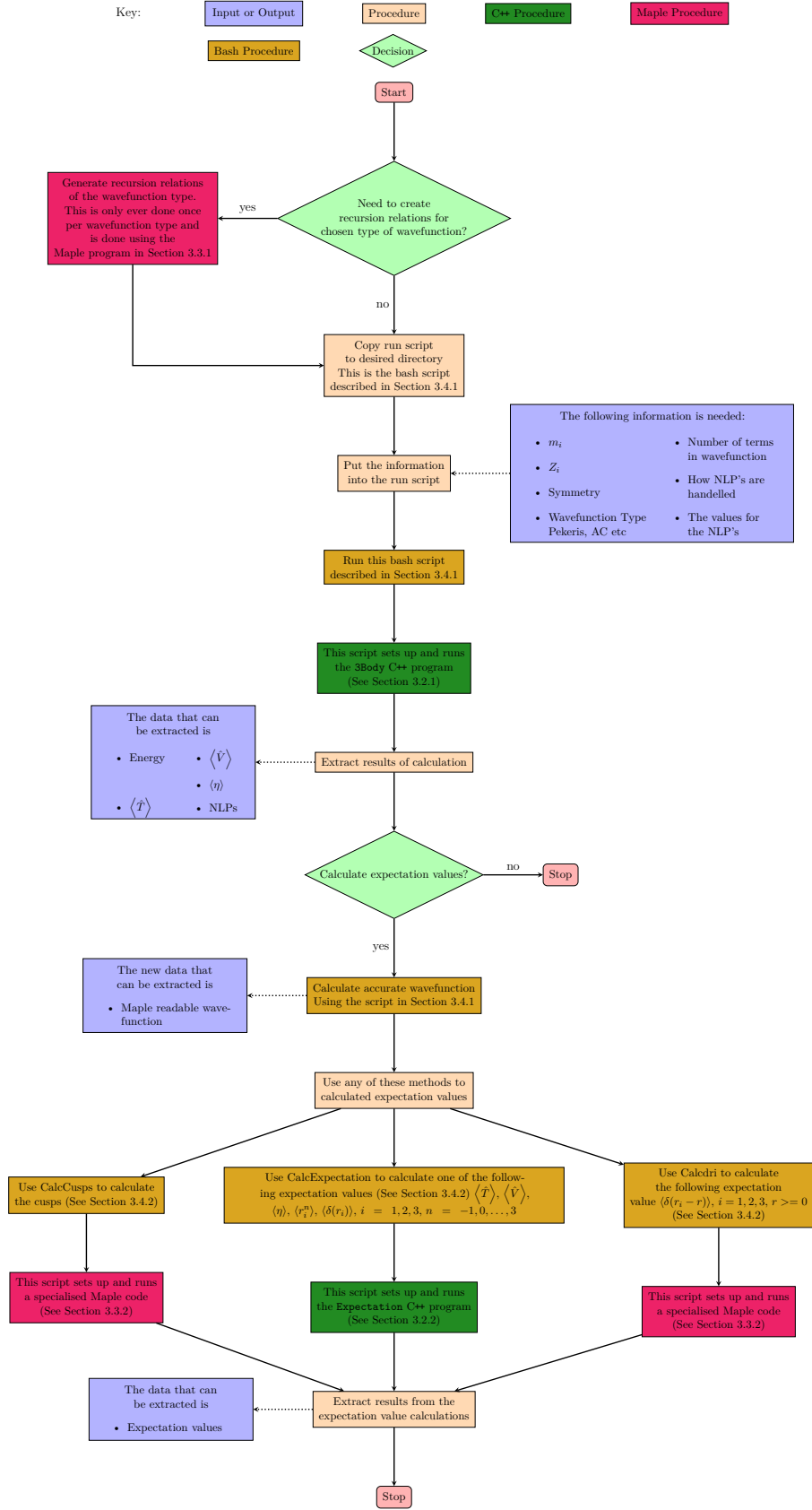
To achieve the design philosophy of not writing the same piece of code twice the **3Body** code is modular. It has two main layers the **3Body** executable and the **Eigen** library. These are the C++ codes and they shall be discussed first. The advantage of this philosophy is best explained by an example. The **3Body** program needed many

### 3. Computer Programming

components to calculate its eigenvalues from a series solution input. These included reading an algebraic series solution recursion relation to create a matrix, and parsing mathematical expressions to numerical values. These two capabilities are identical in requirement to calculating expectation values with the program named **Expectation**. The **Expectation** program also reads a series solution recursion relation to create matrices and parse mathematical expressions to numerical values, for the purpose of calculating the various expectation values given in Section 3.2.2. By writing the separate **Eigen** library for the **3Body** program when writing the **Expectation** program, almost all of the code was already available. The **Expectation** program uses the **Eigen** library and conducts it to do a different task. These two programs share the same code. This means that if a bug is found in the library then fixing that bug fixes it in both programs at the same time, it also reduces the sizes of the programs as they both share the library, only needing one copy of the code between them.

The flowchart in Figure 3.1.1 gives the operational procedure for the three-body programs and codes.

### 3. Computer Programming



**Figure 3.1.1.:** The program operation overview for calculating three-body energies, wavefunctions and expectation values

#### 3.1.1. Programming Language

The language of choice was C++ an object orientated (OO) language and this object orientation was put to good use. It is the scaffolding behind the modularity of the code. Classes, inheritance, polymorphism, templates and specialisation which are aspects of C++ and object orientation were employed. These concepts are explained in the following discussion:

**Class** A class is the main unit of object orientation in C++ it is a collection of methods, variables, operators and other components to perform a specified task. These can be seen as complete units of code for a specific task. When an object is created it conforms to general patterns but can be set up to behave in a variety of ways. For example the class called `Matrix` can be set up to use either mathematical expressions for its elements or numerical values, leading to internal differences between them, but in both cases they have the command `GetElement(x,y)` which will return the element at column x row y. Calling `GetElement` will always give the same output despite the differences between the methods. The classes, where appropriate, have operators defined, for example the `Matrix` class has operators to add and subtract matrices etc. The idea behind a class is to simplify code on a higher level. The programmer need not worry about how a matrix is added just that it will do it when asked and that the class handles all this internally in the appropriate way. An object is a named instance of a class.

##### Class Example

```
1 class AClass{
2     string myText="I am AClass";
3 public:
4     void print(){
5         cout << myText << "\n";
6     }
7 }; //A simple class
8 AClass A; //Create an instance (object) of the class named A
9 A.print(); //Prints the text "I am AClass"
```

### 3. Computer Programming

**Inheritance** Classes can have children and also parents but do not necessarily have either of these. The parent is referred to as the base class and this base class usually contains the general purpose of the class, with default implementations. The children classes referred to as the derived classes will be specialised versions with all the capabilities, and are typically more complex. Take for example a class named **Element** this is the base class and it contains the function **GetValue**. In the base class this simply returns the value it has stored, but the derived classes operate differently. One of these derived classes deals with converting mathematical expressions into a value, when the method **GetValue** is called this class takes different steps, it enters the variables into the expression and evaluates it to a number. Whilst each class handles things differently internally they all have this **GetValue** method which is dictated by the base **Element** class. This base **Element** class guarantees that whatever the derived class does it will at least do this. This leads us on to polymorphism.

#### Inheritance Example

```
1 class AClass{
2     string myText(){
3         return "I am AClass";
4     }
5 public:
6     void print(){
7         cout << myText() << "\n";
8     }
9 };
10 class BClass : public AClass{
11     string myText(){
12         return "I am BClass";
13     }
14 }; //B class is derived from AClass
15 AClass A;
16 A.print(); //Prints the text "I am AClass"
```

### 3. Computer Programming

```
17 BClass B; //Inherits all members, and functions of AClass including
    ↳ print()
18 //As print() calls myText() and myText() has been overridden it
    ↳ instead prints:
19 B.print(); //Prints the text "I am BClass"
```

**Polymorphism** When a `Matrix` class is initiated it is instructed to use one of the `Element` classes but the programmer does not know at compile time which one might get called at run time as it depends upon the users input. The programmer could either create a complicated set of cases to capture all possibility or exploit inheritance and polymorphism. These two work hand in hand and are best employed together. Polymorphism is a complicated name for a simple concept. When a pointer (a reference) to a base class is made it can point to any of its derived classes instead of the base class. For example, if the `Matrix` class is initiated with a derived `Element` class but with a base class pointer, when the pointer is accessed and, for example, calls `GetValue` due to polymorphism the derived class method is called even though the pointer is of the base class type. Any derived class can fit into the pointer and so one pointer can be used to access many variants of the method. As a final note a pointer to a base class cannot access new aspects of the derived class only the parts of the derived class shared with the base can be accessed. There are however techniques to access specific aspects of the derived classes where we recast the pointer by using for example `dynamic_cast`.

#### Polymorphism Example

```
1 class AClass{
2     string myText(){
3         return "I am AClass";
4     }
5 public:
6     void print(){
7         cout << myText() << "\n";
```

```

8     }
9 };
10 class BClass : public AClass{
11     string myText(){
12         return "I am BClass";
13     }
14 }; //B class is derived from AClass
15 class CClass : public AClass{
16     string myText(){
17         return "I am CClass";
18     }
19 }; //C class is derived from AClass
20 AClass A;
21 A.print(); //Prints "I am A Class"
22 AClass B = BClass(); //Not polymorphism because a pointer wasn't
    ↪ used
23 B.print(); //Prints "I am A Class"
24 AClass* APointer = new BClass(); //Polymorphism because a pointer
    ↪ was used
25 APointer->print(); //Prints "I am B Class"
26 delete APointer; //Pointers need to be deleted explicitly
27 APointer = new CClass(); //Pointers can be reassigned
28 APointer->print(); //Prints "I am C Class"

```

**Templates** The C++ language has a technique called templating. In a template class or method certain aspects are left undefined until compile time. In this work we leave the **type** undefined. Therefore our data type could be integer, float, double or even arbitrary precision at a moments notice. This fits in well with the design philosophy of not repeating the same code twice by providing a means to have one source code for all data types. Where possible all code was written in a generic template with the type undefined and then at compilation the types are decided and substituted into the classes. This leaves us with only one set of code and one set of potential bugs to deal with for any arbitrary



### 3. Computer Programming

number of data types.

#### Template Example

```
1 template<typename T>
2 string NumtoStr(const T& Number) {
3     stringstream s; s.str(""); //Create a blank string stream
4     s << setprecision(2) << Number; //Set the stringstream precision
5     ↪ to 2 digits and converts the number to a string
6     return s.str();
7 }
8
9 float mFloat = 1.23;
10 double mDouble = 4.56;
11 int mInt = 7;
12 string numberString = NumtoStr(mFloat);
13 cout << numberString << "\n"; //Prints "1.2"
14 numberString = NumtoStr(mDouble);
15 cout << numberString << "\n"; //Prints "4.5"
16 numberString = NumtoStr(mInt);
17 cout << numberString << "\n"; //Prints "7"
```

**Specialisation** While creating a generic template that is designed to be used for all types it may become necessary to actually be more specific with certain types. For example the function `NumtoStr` converts numbers to strings. The integer, float and double types all convert in the same way using `stringstreams` and the pipe operator, however the arbitrary precision type (`mpreal`, see Section 3.2.1) is special and is converted in a more complex way. Therefore this template type is specialised so that when the function `NumtoStr` is called if the type is the arbitrary precision type a different specialised piece of code is called.

#### Specialisation Example

```
1 template<typename T>
2 string NumtoStr(const T& Number) {
```

### 3. Computer Programming

```
3  stringstream s; s.str(""); //Create a blank string stream
4  s << setprecision(2) << Number; //Set the stringstream precision
   ↪ to 2 digits and converts the number to a string
5  return s.str();
6  }
7  template<>
8  string NumtoStr(const mpreal& Number)
9  {
10     return Number.toString("%.2f") //Converts to string with 2 digits
   ↪ of precision
11 }
12 float mFloat = 1.23;
13 mpreal mMpreal = 4.56;
14 string numberString = NumtoStr(mFloat);
15 cout << numberString << "\n"; //Prints "1.2"
16 numberString = NumtoStr(mMpreal);
17 cout << numberString << "\n"; //Prints "4.5"
```

In addition to C++ we also use Maple and Bash for our computational needs. Maple handles the complex integrals that cannot be solved with series solution in C++ and, the bash scripts handle the logging of the results and simplifies implementation and repetitive tasks. The bash scripts can be considered one of the important parts of the process as they ensure consistency. These scripts are designed to keep a clear record of what has been calculated. This was particularly important in later work as it ensured that original data could be examined and verified a long time after calculation. The Maple program with its graphical user interface (GUI) in contrast is not helpful in this manner. Maple can be run from both the GUI and the shell. In the Maple GUI logs of the calculations are not kept on file and history is lost when the Maple script is restarted. For this reason the procedure has been to create and test the Maple script within the GUI, and when they are completed it is given over to a bash script to run Maple from the shell. The bash scripts are designed to log the results to a file and ensure good records are kept. The C++ programs are also controlled via bash scripts and this removes some of the burdens on the user by

### 3. Computer Programming

helpfully laying out the options of the C++ programs.

Effort has also been devoted to increasing the speed of these programs to that extent parallelisation, the process of breaking a task down to run on multiple cores at the same time, has been employed. In the C++ programs Apple's Grand Central Dispatch<sup>[32]</sup> was used to achieve this as it is optimised for our mac deployment environment. The Grand Central Dispatch is under the Apache License version 2.0 (See Appendix F.1).

#### 3.1.2. Build Environment

The versions of the tools, programs and operating system given in this section are the latest versions confirmed to run the codes written in this work, earlier versions of these tools were used at earlier stages of development. The C++ codes in this project were developed on a Mac Book Pro (15-inch, Mid 2012) with a 2.3 GHz Intel Core i7 CPU and with 16 GB of 1600 MHz DDR3 RAM. The Mac operating system was version 10.10.4 (14E46).

The programs and libraries were compiled for architecture x86-64. The C++ dialect was C++11 with the standard library LLVM C++ with C++11 support. The code was optimised with `-Os` compiler flag that generates the smallest fastest build. The integrated development environment (IDE) xCode was used at version 6.4 (6E35b). The C++ programs were compiled and linked using clang++ Apple LLVM version 6.1.0. Additional post processing of the libraries was achieved using `otool` version `cctools-862`.

Maple 2015.0, Maple build ID 1022128 was used to run the Maple codes. The Bourne-Again shell (bash) shell scripts were run using bash version 4.3.39(1)-release.

## 3.2. C plus plus Programs

There are 3 C++ programs and 1 C++ Library written in this work these are called `3Body`, `Expectation`, `HF2e-Integrals` and `Eigen` respectively. The Library `Eigen` is shared between `3Body` and `Expectation` and is the main work horse performing the most complex steps of the program. A general overview of the library `Eigen` is given but is not discussed in detail. The programs have many options and methods

### 3. Computer Programming

programmed into them, some of which will be briefly discussed. The philosophy of adding code rather than replacing code was taken so as to ensure that if necessary the code could be reverted back to a previous implementation. For example the original code used a self coded mathematical expression to numerical value parser, however this method was slow, inefficient, memory intensive and ultimately replaced with the library `mathpresso` which is fast and more rigorously tested. However the old expression parser is still present and can be re-enabled should the need arise, for example if a future incompatibility with `mathpresso` arises.

#### 3.2.1. 3Body

This program, named `3Body`, calculates the energies, wavefunctions and a few expectation values such as  $\langle \hat{V} \rangle$  and  $\langle \hat{T} \rangle$ . It is a command-line tool which means that it is run from a terminal and a shell. Output is directed to standard out which may either be the terminal itself, a file or another pipe. Like most command-line tools the program is instructed through its arguments in what it should do. An example of running `3Body` with its command-line arguments is as follows:

```
Single Point Calculation
1 3Body Z1=-1 Z2=-1 Z3=1 m1=1 m2=1 m3=1 A=1 C=2 -r RR.rr -s 2856 -SYM
2
```

This calculates the  $\text{Ps}^-$  system with the NLPs  $A$  and  $C$  set to 1 and 2 respectively. The file containing the recursion relations is given with the `-r` option as `RR.rr` (the `3Body` program can also take `-m Matrix.m` to load a matrix calculated in Maple rather than loading the equations to form the matrix, this is however slower), the number of terms in the wavefunction is specified with `-s` option as 2856 and the symmetry is specified as symmetric with the `-SYM` option. The file with the recursion relations (`RR.rr`) is created in Maple, the computer algebra program. This recursion relation file contains the equations to create the matrices in the C++ program. These equations have various undefined variables in this case these are  $Z_1$ ,  $Z_2$ ,  $Z_3$ ,  $m_1$ ,  $m_2$ ,  $m_2$ ,  $A$  and  $C$ . The `3Body` program is designed to read an arbitrary number of variables into the equations. It can take any command-line argument with a “=” as a variable.

### 3. Computer Programming

To perform an optimisation and minimise the non-linear variational parameters we would pass the following arguments into the 3Body program:

```
Optimisation Calculation
1 3Body Z1=-1 Z2=-1 Z3=1 m1=1 m2=1 m3=1 A=opt=1 C=opt=2 -r RR.rr -s
   ↪ 2856 -SYM -bobyqa
2
```

This code will calculate  $\text{Ps}^-$  but will optimise the NLPs  $A$  and  $C$  using the routine BOBYQA.<sup>[33,34]</sup> In Section 3.2.1 alternative optimisation routines are listed and they are controlled by replacing the option `-bobyqa` with the appropriate option for another routine as listed in Section 3.2.1.

The 3Body program itself is relatively simple as most of the code is written in the separate `Eigen` library. We will explain each part of the code that has significance in terms of design or when certain external libraries were used.

#### Precision Options

A C++ executable starts in the main block (`int main(int argc, const char * argv[])`). In our code this main block is short and is a simple gateway to a template function referred to here as the template main block. The template main block has the data type left undefined. It uses a place holder for this data type of `T`. In this template main block rather than writing `double* Vector = new double[size];` it is written as `T* Vector = new T[size];`. The data type `T` is replaced with the appropriate data type for the user requested precision.

The command-line option `-arb NUM` would request `NUM` digits of precision and set `precision` to this number. The main block then decides the data type to use:

```
Setting up arbitrary precision and calling the template main block
1 if (precision >= 20) {
2   SetArbitraryDigits(precision); //Sets up the program to use
   ↪ arbitrary precision
3   main_templated<mpreal>(Args); //calls the templated main block
   ↪ with the type mpreal
4 }
```

```

5 else if (precision >= 16) {
6     SetArbitraryDigits(19);
7     main_templated<long double>(Args); //calls the templated main block
      ↪ with the type long double
8 }
9 else {
10    main_templated<double>(Args); //calls the templated main block
      ↪ with the type double
11 }

```

If less than 16 digits of precision is requested or the command-line argument `-arb` is omitted the data type `double` is used. For greater than 15 and less than 20 `long double` is used. For precisions greater than 19 the arbitrary precision type `mpreal` is used. The template main block is then called with the correct type for the designated precision.

**External Library `mpfrc++`:** The data type `mpreal` is part of the external `mpfrc++`<sup>[35]</sup> library which is a C++ interface for the arbitrary precision data type of the external library `mpfr`.<sup>[36]</sup> The number of digits in this data type are restricted solely by the available memory. The `mpfrc++` library used in this work is version 3.5.6 (Mercurial revision d8d1f40) under the General Public License (See Appendix F.2) and the `mpfr` is version 3.1.2 and is under the lesser General Public License (See Appendix F.3). The `mpfr` library is an excellent general purpose arbitrary precision data type however for certain cases such as 32 and 64 digits there exists more optimised libraries with data types explicitly designed to handle these precisions such as the `qd` library.<sup>[37]</sup> Future work would implement these libraries for increased computational efficiencies.

#### Loading The Matrices

In the template main block the first step creates the matrices. There are 5 matrices in the program. The 3 matrices that represent the operators  $\hat{T}$ ,  $\hat{V}$  and  $\hat{S}$  (see Sections 2.2 and 2.3 on page 7 and on page 8) and two matrices labelled as  $hh$  and  $ss$  that represent the left and right hand side respectively of the generalised

### 3. Computer Programming

eigenvalue problem Eq. (3.2.1).

$$\left(\hat{T} + \hat{V}\right)\psi = E\hat{S}\psi \quad (3.2.1)$$

The recursion relations generated in the Maple program (Section 3.3.1) (and read into this program with the command-line option `-r FILE` or `-m FILE`) contain the mathematical expressions to create all of these matrices. The `3Body` program was designed to have one general matrix class called `Matrix`. This class would contain these mathematical expressions and would generate upon request any of the 5 matrices ( $\hat{T}$ ,  $\hat{V}$ ,  $\hat{S}$ ,  $hh$  or  $ss$ ). This is the first example of the use of inheritance and polymorphism. The base class `Matrix` is designed to return any of the matrices using the function `GetMatrix(MatrixPart Part)` where `Part` tells it to return for example the  $hh$  matrix. How this is done however depends on which derived class of the `Matrix` class was used. The class `Matrix_RR_JIT` for example converts the recursion relations to numerical numbers using the external library `mathpresso`<sup>[38]</sup> coupled with just in time (JIT) compilation of the mathematical expression to assembled machine code (ASM) for maximum speed. Whereas the class `Matrix_RR_NOJIT` does it without JIT compilation which is slower but the only option for data types other than `double`.

The class `Matrix` and its derived class handle the conversion of the mathematical expressions to numerical data. All of the appropriate work is handled by the `Eigen` library and all this program needs to do is create the appropriate class for the desired method of solving these expressions. By default this is `Matrix_RR_JIT` for the command-line argument `-r` and `Matrix_JIT` for the command-line argument `-m`. To use the non-JIT matrix class the command-line options `-NOJIT` is supplied. The following is an example of its use:

An example Use of the Matrix class and its Derived Classes

```
1 Matrix<T> *TheMatrix = nullptr;
2 VariableSet<string> InitVars;
3 InitVars.Add(Variable<string>("m1", "1"));
4 InitVars.Add(Variable<string>("m2", "1")); //and so on
```

### 3. Computer Programming

```
5 LMN::SymmType Symm = LMN::SymmType::Symmetric; //The wavefunction
   ↳ symmetry. This is needed for the recursions relations
6 MatrixRules* Rulesp = nullptr; //Rules for loading the Matrix parts
   ↳ KE, PE, OV, etc from the Matrix class i.e. what markers are used
   ↳ (See Section 3.3.1)
7 KERHS=false;
8 vector<string> WithT{"t"};
9 vector<string> WithoutT(0);
10 vector<string> WithV{"p"};
11 vector<string> WithoutV(0);
12 vector<string> WithO{"ss"};
13 vector<string> WithoutO(0);
14 Rulesp = new
   ↳ MatrixRules(WithT,WithoutT,WithV,WithoutV,WithO,WithoutO, KERHS);
15 long RR_MatSize = 1078;
16 TheMatrix = new Matrix_RR_NOJIT<T>(FileName, RR_MatSize, Symm, Rules,
   ↳ InitVars); //Initialise the matrix to explicitly use the
   ↳ recursion relation of the Laguerre polynomials (faster but
   ↳ specialised to a Laguerre polynomial wavefunction)
17 //or
18 TheMatrix = new Matrix_MathJIT<T>(FileName,-1, Rules, InitVars);
   ↳ //Initialise the matrix to read .m file without regard for the
   ↳ recursion relations (slower but more general allowing any
   ↳ eigenproblem to be diagonalised). A size of -1 is passed to read
   ↳ the size from the file
19 VariableSet<string> Vars; //We can later add or change variables in
   ↳ the following way
20 Vars.Add(Variable<string>("A","1"));
21 Vars.Add(Variable<string>("C","2"));
22 TheMatrix->EnterVariables(VariableSet<string> Vars);
23 TheMatrix->GetMatrix(MatrixPart::HH)->Print(); //prints the HH matrix
   ↳ to the terminal.
```



```
24 TheMatrix->GetMatrix(MatrixPart::SS)->Print(); //prints the SS matrix  
↪ to the terminal.
```

**External Library `mathpresso`:** The library `mathpresso`,<sup>[38]</sup> or more specifically the fork `doublepresso` (revision d4a8eb3) is an external library for evaluating algebraic mathematical expressions with a set of variables to a numerical value. This external library is under the MIT license (See Appendix F.4). The library depends upon `ASMJit` version 1.0 which is under the `zlib` license (See Appendix F.5). The `mathpresso` library was chosen as it also incorporates JIT compilation to ASM. This means that when the recursion relations are initialised in a `mathpresso` object the mathematical expressions are converted into machine code this is a much faster way of calculating these expressions then parsing the expression as tokens or strings. When the `3Body` program was first created I wrote a simple class as a parser to evaluate the mathematical expressions called `MathString`. The `mathpresso` equivalent was several times faster and a much more developed library. This library is however not optimised for arbitrary precision. Future work could be directed to using the shunting yard algorithm<sup>[39]</sup> to create a mathematical parser optimised to work with arbitrary precision and our eigenproblem.

#### Choosing the `EigenSolver`

In the next step the C++ program `3Body` selects and prepares the algorithm to diagonalise the matrices and solve the general eigenproblem. Two algorithms have been implemented in this code the QL algorithm (equally the QR algorithm could have been used which uses the upper rather than lower quartile of the matrix but is functionally equivalent as the algorithm requires a symmetric matrix) and the multiple relatively robust representations (MR3) algorithm. The MR3 algorithm is currently the fastest algorithm for symmetric matrices and was formulated in 2004. The `3Body` program has the base class `EigenSolver` this defines common methods such as Cholesky Decomposition<sup>[29]</sup> and House Holder reduction.<sup>[29]</sup> An understanding of these algorithms was derived from the Numerical Recipes books<sup>[29,30]</sup> but were adapted to be able to run in parallel and to use templates and arbitrary precision.

### 3. Computer Programming

The Cholesky decomposition needs to have an overlap matrix matrix that is positive definite. The Cholesky decomposition is not particularly well suited for a parallel method due to the dependency order of the calculation steps. There is research in the literature that attempts to make the Cholesky decomposition algorithm more suited for parallel calculation that may be employed in future work.<sup>[40,41]</sup>

The two derived classes `QL` and `MRRR` operate in a relatively simple manner. At initialisation it takes a reference to a `Matrix` object and then on demand it takes a set of variables to enter into this matrix and solve the general eigenproblem. It uses the function `GetMatrix(MatrixPart Part)` with `Part` equal to `MatrixPart::HH` and `MatrixPart::SS` to generate the appropriate matrices to solve the eigenvalue problem. The class `QL` implements the QL algorithm of Numerical Recipes.<sup>[29,30]</sup> The class `MRRR` uses the MR3 algorithm provided by `lapack` using the accelerate framework's bindings.<sup>[42]</sup> Both the eigenvectors and the eigenvalues can be calculated by the program. The QL algorithm will calculate all eigenvectors or none whilst the MR3 can calculate specific eigenvectors as requested. The following is an example of how to use these classes.

#### Examples uses of the EigenSolver class

```
1 EigenSolver<T>* TheDiagonaliser = nullptr;
2 TheDiagonaliser = new QL<T>(*TheMatrix); //For the QL method
3 \\or
4 TheDiagonaliser = new MRRR<T>(*TheMatrix); //For the MR3 method
5 ep2E<T> *EVConverter; //A class that tells the program how to
   ↳ convert the eigenvalue ( $\lambda$ ) to the desired final result
6 EVConverter = new MassHamiltonian<T>(); //For the mass Hamiltonian
   ↳ this is  $m_2 = \frac{-1}{\lambda}$ 
7 EVConverter = new ChargeHamiltonian<T>(); //For the charge
   ↳ Hamiltonian this is  $Z_3 = \frac{-1}{\lambda}$ 
8 EVConverter= new ep2E<T>(); //For the energy Hamiltonian this is
   ↳  $E = -\lambda$ 
9 TheDiagonaliser->SetEVConverter(EVConverter);
```

### 3. Computer Programming

```
10 TheDiagonaliser->SetRoot(1); //We can ask the diagonaliser to give
    ↳ us a specific root of the equation however in this thesis we only
    ↳ used the first root. However the program is capable of looking at
    ↳ any of the roots with the -root  $X$  option to study the  $X^{\text{th}}$  root.
11 VariableSet<string> Vars;
12 Vars.Add(Variable<string>("A","1"));
13 Vars.Add(Variable<string>("C","2"));
14 bool CalculateEigenVectors=true; //The diagonaliser is set to
    ↳ calculate the eigenvectors too
15 long NumberofEigenVectors=1; //Only calculate the first
    ↳ eigenvector (if possible)
16 EigenResults<T> Results = TheDiagonaliser-
    ↳ >Solve(Vars,CalculateEigenVectors,NumberofEigenVectors);
    ↳ //Calculate eigenvalues and save the results to EigenResults<T>
    ↳ Results.
17 cout << "The eigenvalue was " << Results.LowestEigen().EigenValue <<
    ↳ " with Energy of " << Results.LowestEigen().Energy << "\n";
    ↳ //Print the results
```

**External Library Numerical Recipes:** The numerical recipes code comes from Numerical Recipes in C (1<sup>st</sup> edition)<sup>[29]</sup> and Numerical Recipes in C++ (3<sup>rd</sup> edition).<sup>[30]</sup> These were useful reference materials as they detailed the general processes that would be needed in writing the numerical Eigen solver classes. The Numerical Recipes code is under a restrictive license which shall be referred to as the NR License (See Appendix F.6).

**External Library Lapack:** The lapack routines are perhaps one of the best known Eigen solver libraries. Written in Fortran 90 and ported into various other languages, they provide efficient and effective routines to solve many linear algebra problems including simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. lapack uses BLAS (Basic Linear Algebra Systems) a set of routines for efficient matrix operations.

### 3. Computer Programming

In some cases manufacturers of machines include optimised variants of BLAS for their machines, these however were not used in this work but may be good future work. The MR3 algorithm was called through the function `dsyevr`. This function was exposed using the accelerate frameworks bindings. The accelerate framework is a Mac only propitiatory framework and is only available on Mac OS. However in this work only the bindings to `lapack` are used this could therefore be replaced with a direct link to `lapack` or better still `mpack`<sup>[43]</sup> which is a C++ port that allows for the use of arbitrary precision. This is however not a small task but is considered as good future work. The lapack code is under a modified BSD license (see Appendix F.7).

#### Parameter Optimisation

There are 3 primary ways to find the optimum value of the NLPs:

**Scan** Scanning the NLPs involves calculating a series of equispaced points between a range of  $a$  to  $b$  in steps of  $s$ . Where  $b > a$  and  $s < |b - a|$ . This optimisation is implemented by the command-line argument `A=scan=a,b,s`.

**Optimise** When referring to optimising the NLPs this is the option being referred to. The local minimum (or upon request the maximum) is located starting from an initial guess value. In this work the guess values are always the formal asymptotic solution (see Section 4.2.7 on page 100). Optimisation is performed using techniques such as quasi-Newton-Rapson and conjugate gradients. The optimisation is implemented by the command-line argument `A=opt=startvalue -OPTALGORITHM -tol OPTTOLERANCE`. The optimisation routines came from external sources such as `dlib`<sup>[44]</sup> and numerical recipes.<sup>[29,30]</sup> Each different algorithm will minimise the system in a different way and even their end condition are not necessarily comparable. In most cases these algorithm's end condition is to reduce the gradient to below a specified tolerance or reduce the size of the trust region to a specified size.

**Genetic Algorithms** Genetic algorithms have been implemented to calculate global minimum in a range between  $a$  and  $b$  where  $b > a$ . This is done using the external library `GAlib` which has an extensive list of genetic algorithms and techniques. It is also a very well documented and modular object orientated

### 3. Computer Programming

piece of code. The command-line argument is `A=GA=a,b -pop POPSIZE -ga GENETICALGORITHM`

To facilitate these different methods to find the minimum the program uses inheritance and polymorphism. A base class `ParameterSearcher` with a simple set of general functions and properties was written. The actual work of optimisation is handled by its derived classes `Scan`, `Optimise` and `GA_Search`. When the derived classes are created they are set up with all necessary information unique to their implementation. The `Optimise` and Genetic Algorithm (GA) methods require some more in-depth explanation of their available options and their effects upon the calculations.

**Optimise** The `Optimise` methods include the following algorithms sorted alphanumerically by their command-line argument that activates them.

**-bfgs** The Broydon-Fletcher-Goldfarb-Shanno algorithm.<sup>[45,46]</sup> The algorithm is provided from the external `dlib` library and is a quasi-Newton-Rapson method for finding unconstrained minimum. Quasi-Newton-Rapson methods unlike Newton-Rapson methods circumvent the need for second derivatives by approximating the second derivatives.

**-bobyqa** The Bound Optimisation BY Quadratic Approximation algorithm is an algorithm written by Powell in 2009.<sup>[33,34]</sup> This algorithm is provided from the external `dlib` library. This algorithm was found to be our most effective method, capable of isolating a minimum more effectively than other methods and in a shorter time. The method does not need any gradient information, but must be given finite bounds in which the minimum will be located in addition to an initial guess value. The bounds implemented in this program are  $1 \times 10^{-6}$  to  $1 \times 10^4$ . A range that is almost certain to contain the minimum. Later this optimisation method is discussed more deeply along with how it can be combined with our second best algorithm `-NROpt` (Section 3.4.1).

**-cg** The Conjugate Gradient algorithm.<sup>[29]</sup> There are three different implementations of the conjugate gradient algorithm in this program. This algorithm is also provided by the external `dlib` library but this method unlike the other two

### 3. Computer Programming

algorithms was found to be the least effective being both slow and poor at localising a minimum for reasons discussed later.

**-lbfgs** The limited-memory Broydon-Fletcher-Goldfarb-Shanno algorithm.<sup>[47]</sup> This method approximates the Broydon-Fletcher-Goldfarb-Shanno algorithm and is provided from the external `dlib` library. For a problem with many NLPs ranging in the hundreds the full Broydon-Fletcher-Goldfarb-Shanno algorithm would have a large memory requirement as it stores a large  $n \times n$  approximate Hessian matrix. This limited memory form instead has a much smaller memory requirement. However typically the programs created and used in this work have 2 NLPs and at most only 3 NLPs there has therefore never been need to implement this optimisation routine other than to test its effectiveness.

**-MacOpt** This is a variant of a numerical recipes conjugate gradient routine written by David MacKay.<sup>[48]</sup> This variant attempts to use the gradient information more effectively to guess the step size of the conjugate gradient routine of numerical recipes and also use the gradient information to more efficiently bracket the minimum. The method relies on gradients being more easily calculated than function evaluations.

**-NROpt** This was the first conjugate gradient routine implemented in this work and it was implemented from numerical recipes C++ third edition 2007.<sup>[29,30]</sup> The algorithm proved to be effective at finding minimum and will later be shown how it was combined with the bound optimisation by quadratic approximation (BOBYQA) algorithm to effectively locate minimum.

These algorithms differ in their effectiveness at locating the minimum. To identify the most effective method a series of tests were performed. These tests were carried out on a small 95 term wavefunction of  $\text{Ps}^-$  and  $\text{H}_2^+$ . A small wavefunction was used to help improve the accuracy of the timing data and reduce computational expense during testing. The run times presented here averaged over 15 runs (Tables 3.2.1-a and 3.2.1-b) are subject to change depending on the computer operating it and the computers condition, and should only be taken as a general guide. The optimisation tolerance in all of these was set to  $3 \times 10^{-7}$ .

### 3. Computer Programming

Algorithm	NLP A	NLP C	Energy	Run Time
BOBYQA	0.346 093 2	0.711 393 8	−0.262 004 670 746	1 s, 538 ms
MacOpt	0.346 359 4	0.711 349 5	−0.262 004 670 736	1 s, 838 ms
NROpt	0.346 353 2	0.709 056 1	−0.262 004 670 721	3 s, 552 ms
LBFGS	0.340 073 6	0.974 487 9	−0.262 004 471 026	5 s, 729 ms
BFGS	0.338 822 8	0.976 328 6	−0.262 004 469 284	5 s, 757 ms
CG	0.435 652 7	0.990 426 0	−0.262 002 299 869	1 m, 19 s, 238 ms

**Table 3.2.1-a.:** Results of the different optimisation methods sorted by energy. The calculation run time is averaged over 15 runs. These are the calculations on a 95 term  $\text{Ps}^-$  wavefunction.

Algorithm	NLP A	NLP C	Energy	Run Time
NROpt	4.023 358	1.459 111	−0.595 748 682 180	1 s, 784 ms
MacOpt	4.023 505	1.459 279	−0.595 748 682 162	1 s, 557 ms
LBFGS	2.993 662	1.584 131	−0.595 178 850 978	1 s, 397 ms
BFGS	2.993 672	1.583 577	−0.595 178 850 576	1 s, 944 ms
CG	1.471 473	1.866 864	−0.591 613 585 358	2 m, 395 ms
BOBYQA	0.933 022 9	2.157 164	−0.586 794 255 364	1 s, 143 ms

**Table 3.2.1-b.:** Results of the different optimisation methods sorted by energy. The calculation run time is averaged over 15 runs. These are the calculations on a 95 term  $\text{H}_2^+$  wavefunction.

The energies of Tables 3.2.1-a and 3.2.1-b show that in both  $\text{Ps}^-$  and  $\text{H}_2^+$  LBFGS, BFGS and CG perform less well than the others with the exception of BOBYQA which performed well with  $\text{Ps}^-$  and not with  $\text{H}_2^+$  this will be explain shortly. The numerical recipes algorithm (NROpt) and more gradient reliant counter part (MacOpt) both perform reasonably similarly with the exception that MacOpt is typically faster. This is because MacOpt uses the gradient information to perform less optimisation steps so that even though each gradient request is expensive this is compensated by its improved efficiency. However we found that MacOpt would sometimes take large steps in the wrong direction during its calculation which made NROpt our preferred choice (see Figures 3.2.1b and 3.2.1c). BOBYQA was always the best choice for atomic systems.

Figure 3.2.1 show the path each algorithm takes to localise its minimum. NROpt (Figure 3.2.1b, MacOpt (Figure 3.2.1c) and BOBYQA (Figure 3.2.1g) all isolate the minima successfully. However the other algorithms fail to make head way and all give up on finding the minimum fairly quickly. To understand why observe the final gradients at the end points of the algorithms given in Table 3.2.2. The algorithms

### 3. Computer Programming

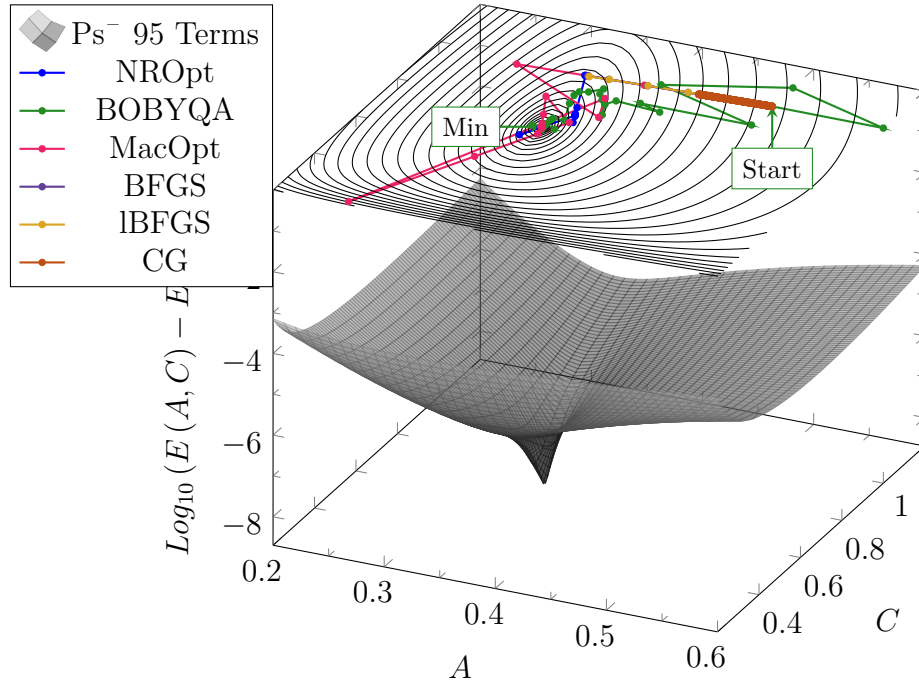
Algorithm	$\frac{\partial E}{\partial A}$	$\frac{\partial E}{\partial C}$
NROpt	$1.721 \times 10^{-7}$	$-3.152 \times 10^{-8}$
MacOpt	$6.453 \times 10^{-8}$	$4.190 \times 10^{-8}$
BFGS	$1.660 \times 10^{-6}$	$-2.620 \times 10^{-1}$
LBFGS	$1.667 \times 10^{-6}$	$-2.620 \times 10^{-1}$
CG	$7.936 \times 10^{-6}$	$-2.620 \times 10^{-1}$
BOBYQA	$6.458 \times 10^{-8}$	$-3.142 \times 10^{-8}$

**Table 3.2.2.:** The final gradients after optimisation for each algorithm with the 95 term  $\text{Ps}^-$  wavefunction. The optimisation tolerance was  $3 \times 10^{-7}$ . BOBYQA algorithm does not require gradients to operate and so has had its gradients calculated especially.

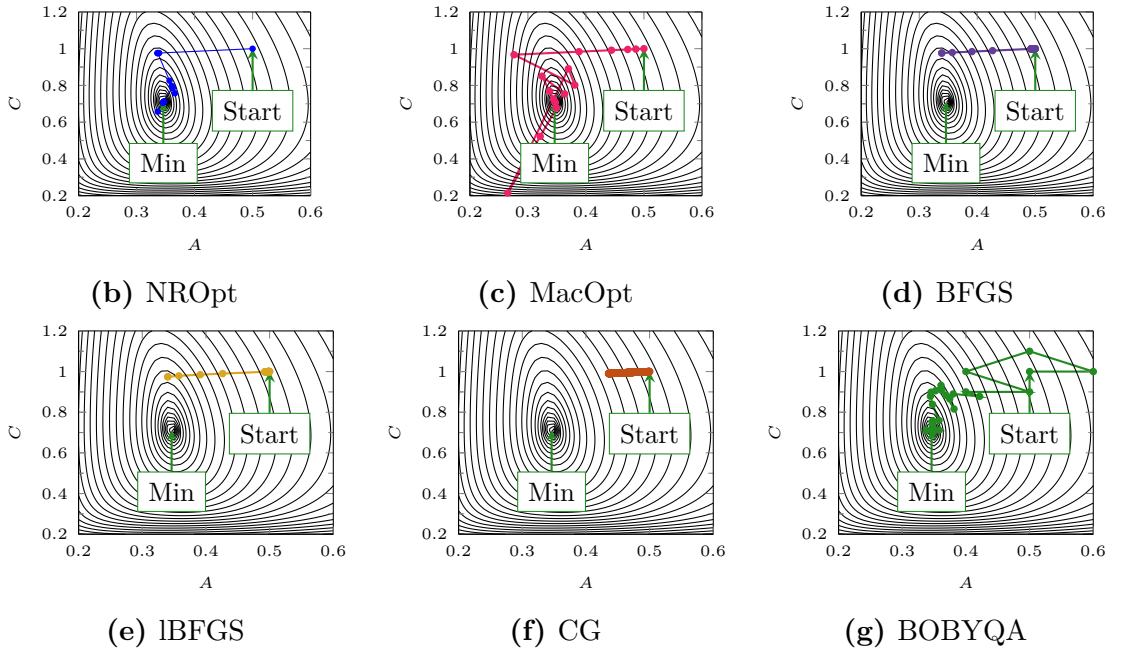
BFGS, LBFGS and CG as provided by the `dlib` library are only minimising one of the coordinates gradient to less than the tolerance before quitting. This means that the end condition is different to the others in that only one gradient is lowered below the tolerance. In the NROpt and MacOpt algorithms they minimise the size of the gradient in both directions to below the tolerance and so their end condition was not achieved until later. The BOBYQA algorithm does not use gradients but uses the trust region size as its end condition. It would not be a difficult task to edit the source code of the algorithms BFGS, LBFGS and CG algorithms in the `dlib` library to use a more suitable end condition but it was decided to instead move forward with currently working algorithms NROpt, MacOpt and BOBYQA. Future work may wish to implement these other algorithms with the correct end condition and identify how they perform.



### 3. Computer Programming



(a) All algorithms on a single plot of  $\text{Ps}^-$ , the z axis of the surface has been adjusted with the transformation  $\text{Log}_{10}(E(A, C) - E_{\min})$  where  $E_{\min}$  is the lowest calculated energy (from BOBYQA). This adjustment enables the minimum of the surface to be more easily seen as otherwise the surface appears flat.



**Figure 3.2.1.:** The optimisation paths taken by the various algorithms, each node denotes a calculation. The contour and surface were created using scan data of increments 0.01, with the optimisation paths superimposed. All data uses the 95 term  $\text{Ps}^-$  wavefunction.

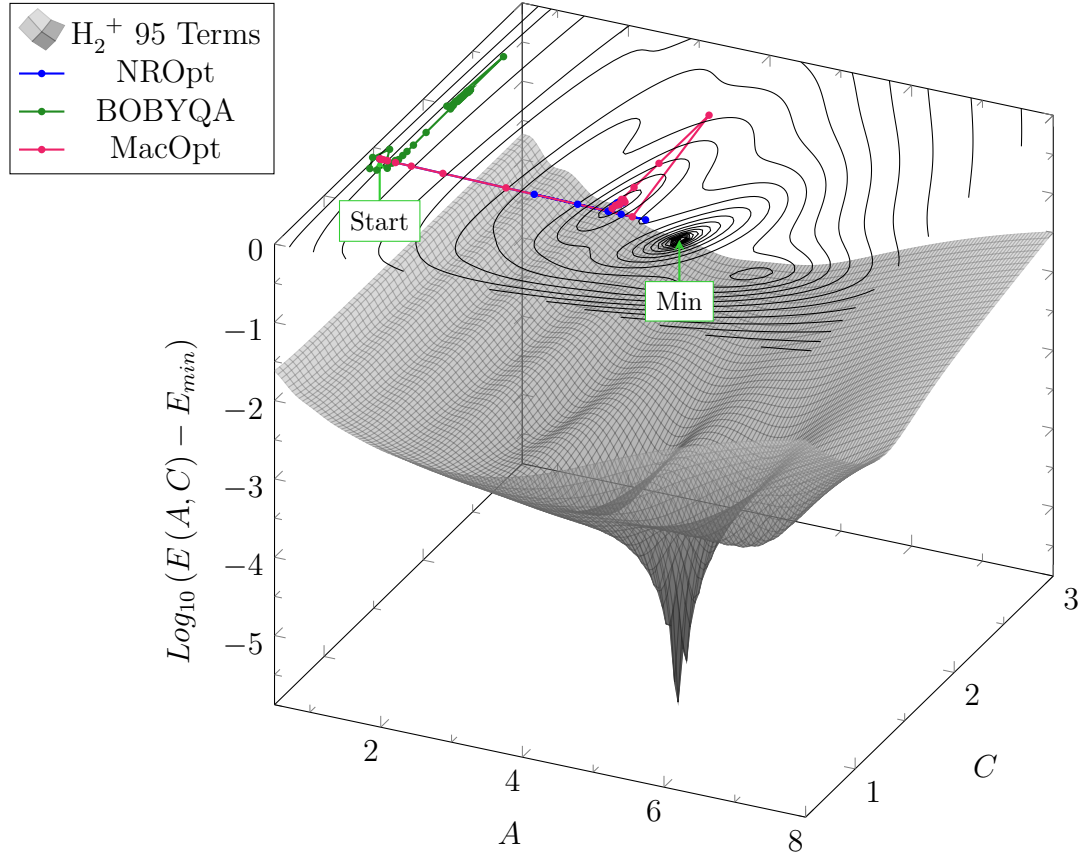
### 3. Computer Programming

The  $\text{H}_2^+$  case is more complex. As is shown in Figure 3.2.2a the molecular  $\text{H}_2^+$  has multiple minima. It was discovered that this is the case with all molecular systems in this work and therefore a greater deal of effort is put into minimising the NLPs of molecular systems as discussed in Section 4.4.1 on page 110. The algorithms BFGS, LBFGS and CG suffer from the same problems in their end condition and so are not investigated on the  $\text{H}_2^+$  surface. The BOBYQA algorithm behaves differently to the other algorithms when confronted with multiple minima and for this reason it has a less stable energy for molecular cases than the other algorithms. In Figure 3.2.2 the steps each algorithm follows is shown.

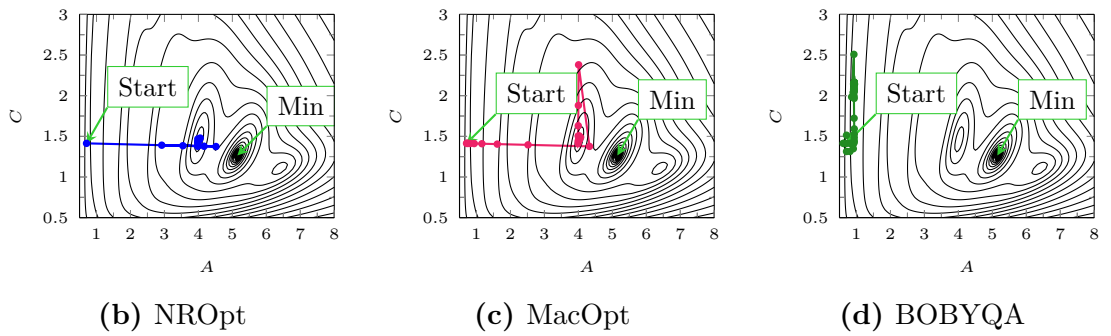
From the algorithm calculation paths in Figure 3.2.2 it can be seen that, for  $\text{H}_2^+$ , BOBYQA fails to locate the lowest minima, this is in contrast to the results of  $\text{Ps}^-$  where BOBYQA was the most effective at finding the lowest value. BOBYQA will typically fall into the nearest minimum but will fall deeper into this minimum than the other methods. For this reason BOBYQA in Table 3.2.1-b gave the least stable energy due to the multi minima  $\text{H}_2^+$  surface but gave the best energy with the single minimum surface of  $\text{Ps}^-$  (Table 3.2.1-a).

The algorithms NROpt and MacOpt do considerably better than BOBYQA at identifying a more stable minimum but they do not reach the lowest minimum. For this reason in Chapter 4 a combination of scan and optimisation was used to achieve the optimum NLPs values for molecular systems. Scans however are inefficient and require a very large number of data points. For example the scan surface in Figure 3.2.2a required 70 000 calculations which is infeasible for wavefunction with many terms.

### 3. Computer Programming



(a) Selected optimisation algorithms on a single plot for  $\text{H}_2^+$ , the z axis of the surface has been adjusted with the transformation  $\text{Log}_{10}(E(A, C) - E_{min})$  where  $E_{min}$  is the lowest calculated energy. This adjustment enables the minimum of the surface to be more easily seen as otherwise the surface appears flat.



**Figure 3.2.2.:** The optimisation paths taken by the various algorithms, each node denotes a calculation. The contour and surface were created using scan data of increments 0.01, with the optimisation paths superimposed. All data uses the 95 term  $\text{H}_2^+$  wavefunction.

### 3. Computer Programming

The following is an example of the code used to perform an optimisation with these local minimum methods. The internal workings of `Optimise` makes further use of polymorphism to select individual algorithms.

#### An example use of the Optimise Class

```
1 ParameterSearcher<T> *Method; //Create pointer to base class
  ↳ Parameter searcher
2 long tol = -3; //Set tolerance to  $3 \times 10^{-3}$ 
3 VariableSet<string> InitVars;
4 InitVars.Add(Variable<string>("m1","1"));
5 InitVars.Add(Variable<string>("m2","1")); //and so on
6 OptVarSet<T> OptVars; //An object to hold the variables to be
  ↳ optimised. This object keeps track of initial, current and final
  ↳ values of the non-linear variational parameters
7 OptVars.Add(OptVar<T>("A",1)); //Variable A is setup to optimise with
  ↳ an initial value of 1
8 OptVars.Add(OptVar<T>("C",2));
9 Method = new Optimise<T>(InitVars,OptVars,TheDiagonaliser,tol);
  ↳ //Create the derived class Optimise
10 Optimise<T>* OptMethod = dynamic_cast<Optimise<T>*>(Method);
  ↳ //Dynamic cast the pointer to the Optimise class so that we can
  ↳ access Optimise specific functions not available in the base
  ↳ class ParameterSearcher
11 OptMethod->SetOptimisationMethod(NROpt); //Set the optimisation to
  ↳ NROpt through the Optimise specific function
12 //or
13 OptMethod->SetOptimisationMethod(MacOpt); //Set the optimisation to
  ↳ MacOpt through the Optimise specific function
14 //or
15 OptMethod->SetOptimisationMethod(dLib_bobyqa); //Set the
  ↳ optimisation to BOBYQA through the Optimise specific function
16 Method->Maximise(false); //Call this function to set it to maximise
  ↳ of minimize
```

### 3. Computer Programming

```
17 Method->TestEnergy(true); //By default the program will minimise the
    ↳ Eigen parameter  $\epsilon$  but if this function is called with true it
    ↳ will optimise the result of applying the conversion from  $\epsilon$  to  $E$ 
    ↳ according to the ep2E<T> class specified previously in the
    ↳ EigenSolver class
18 EigenResults<T> Results = Method->Process(); //This function will do
    ↳ all of the necessary steps to optimise the non-linear variational
    ↳ parameters. Calling the appropriate optimisation algorithms and
    ↳ calling TheDiagonaliser->Solve to solve the eigenproblem and
    ↳ calculate numerical gradients if necessary
19 std::cout << "The eigenvalue was " <<
    ↳ Results.LowestEigen().EigenValue << " with Energy of " <<
    ↳ Results.LowestEigen().Energy << "\n"; //Print the results
```

**External Library dlib:** The library dlib is a general purpose open source C++ library. It is a collection of independent software components, and includes codes on optimisation, networking, threads, linear algebra and machine learning. In this work the numerical optimisation codes of dlib are used, particularly the BOBYQA code. The code is well documented and uses object orientation. I modified the BOBYQA algorithm into a template class for use with arbitrary precision. The dlib library is under the Boost Software License (See Appendix F.8).

**External Library Numerical Recipes:** To implement the NROpt algorithm Numerical Recipes in C (1<sup>st</sup> edition)<sup>[29]</sup> and Numerical Recipes in C++ (3<sup>rd</sup> edition)<sup>[30]</sup> were used. The codes in these books, although written in a more procedural fashion than an object orientated, have useful discussions as to function and design of the procedures. The NROpt method refers to their Polax-Ribiere conjugate gradient algorithm named as `void frprmn(...)` in the book. The algorithms in the book were not copied exactly but rather adapted to make use of arbitrary precision through the implementation of template classes and functions. This code is under a restrictive License which shall be referred to as the NR License (See Appendix F.6).

### 3. Computer Programming

**External Library Mackay Optimisation:** The optimisation algorithm MacOpt was written by David Mackay in 2004. The algorithm was designed to make up for some deficiencies found in earlier Numerical Recipes conjugate gradient algorithms. Including, using the gradient to determine the initial step size of the line minimiser algorithm; using gradients in the line minimiser algorithm so that only two calculations are required to bracket a minimum; using interpolation once a minimum has been bracketed to reduce computation costs. The code was modified to use arbitrary precision with template classes. This code is under the lesser GPL license (see Appendix F.9).

**Genetic Algorithms** The GA were implemented to find the global minimum in the NLPs surface without having to perform many thousands of calculations to describe the surface. The Genetic Algorithms work by first seeding the surface with a population of  $N_{pop}$  size which is done in a uniform fashion using the `GAlib` function `UniformInitializer(GAGenome & c)`. The population then goes through successive generations where the best individuals are picked out and used to form the next generation. How each population evolves over generations depends upon the algorithm employed. In this work the best individuals are picked from a normalised fitness score (using energy, where lower energy means fitter) using the `GAlib` class `GATournamentSelector`. This selector picks two fitness proportional random individuals and then picks the better of the two as the victor. Random mutations are incorporated through the function `int GARealGaussianMutator(GAGenome& g, double pmut)` this adds a random mutation based on a Gaussian distribution centred on its current value.

There are several different Genetic Algorithms available in the `GAlib` library: simple, steady state, incremental, crowding and DEEM. These are all selected with the command-line arguments `-ga Simple`, `-ga SteadyState`, `-ga Incremental`, `-ga Crowding` or `-ga DEEM` respectively. The following is a brief overview of each algorithm as implemented in `GAlib`.

**Simple** This algorithm behaves as a traditional GA.<sup>[49]</sup> It takes the best individuals and breeds them creating a whole new population. The old population completely replaced.

### 3. Computer Programming

**SteadyState** This behaves in the same manner as Simple except that only part of the population is replaced in the **Eigen** library this proportion is set to 60 %.

**Incremental** Is a slow evolution, only the best individuals will breed and they will replace the worse. Typically only two individuals are created per generation. This is a slow evolution and will typically converge too soon for our purposes.

**Crowding** works in a similar way to steady state however rather than replacing the worse individual it replaces the closest parent. It is designed to keep different individuals unique and maintain diversity.

**DEME** This is the most complex algorithm it attempts to maintain multiple SteadyState populations. These populations mix once per generation where they exchange the best individuals. This method typically produces better results but requires a very large number of individuals and is by far the most computationally expensive.

To discuss effectiveness of the GA the  $\text{H}_2^+$  surface is used in this discussion as this surface has multiple minimum rather than the single minimum atomic surfaces as these are easily solved with the local minimum optimisation routines of NROpt and BOBYQA. Table 3.2.3 gives a summary for the GA on the  $\text{H}_2^+$  surface it is sorted in order of the most stable to least stable energy and includes the population size, Genetic Algorithm and the optimisation method results.

From Figure 3.2.2 the ideal minimum is at  $A \approx 5.15$  and  $C \approx 1.26$  the optimisation routines failed to locate this minima and so a successful GA will be one that locates this minimum successfully without taking significantly longer than the optimisation routines. With the exception of the Incremental algorithm most GA successfully locate the minimum. The slowest algorithm was the DEME algorithm as it takes the largest number of calculations to maintain its multiple populations. The Simple algorithm with a larger population size performs well. It is worth noting that due to the random nature of these algorithms the best algorithm is subject to the initial population seed and the mutations that occur. Due to this random nature, greater population sizes are not always better, especially when the energy begins to converge.

**External Library GALib:** The library **GALib** is a comprehensive piece of code

### 3. Computer Programming

Algorithm	$N_{pop}$	A	C	Energy	Time
Simple	150	5.150	1.272	-0.595 877 138	39 s, 328 ms
DEME	20	5.135	1.267	-0.595 876 959	59 s, 595 ms
DEME	10	5.130	1.254	-0.595 876 887	1 m, 2 s, 701 ms
Crowding	100	5.131	1.265	-0.595 876 884	30 s, 841 ms
Simple	200	5.162	1.281	-0.595 876 587	52 s, 542 ms
Simple	100	5.172	1.286	-0.595 876 011	26 s, 786 ms
Crowding	200	5.195	1.280	-0.595 874 885	45 s, 295 ms
Simple	20	5.203	1.292	-0.595 873 995	5 s, 162 ms
Simple	50	5.193	1.301	-0.595 873 476	12 s, 854 ms
DEME	150	5.256	1.316	-0.595 865 066	1 m, 33 s, 758 ms
DEME	200	5.245	1.329	-0.595 864 686	1 m, 44 s, 399 ms
Simple	10	5.261	1.336	-0.595 861 127	2 s, 948 ms
Crowding	150	5.275	1.278	-0.595 854 707	46 s, 616 ms
DEME	100	5.280	1.282	-0.595 853 972	1 m, 21 s, 785 ms
SteadyState	50	5.321	1.328	-0.595 847 627	8 s, 402 ms
Crowding	20	5.175	1.357	-0.595 847 160	6 s, 89 ms
SteadyState	200	5.261	1.384	-0.595 841 659	33 s, 404 ms
DEME	50	5.302	1.403	-0.595 831 245	1 m, 12 s, 27 ms
SteadyState	20	5.001	1.136	-0.595 823 400	3 s, 347 ms
Incremental	20	5.266	1.444	-0.595 797 564	1 s, 110 ms
Crowding	50	4.828	1.178	-0.595 776 889	15 s, 408 ms
NROpt	N/A	4.023	1.459	-0.595 748 682	1 s, 882 ms
MacOpt	N/A	4.024	1.459	-0.595 748 682	1 s, 882 ms
SteadyState	150	3.992	1.420	-0.595 746 469	2 s, 135 ms
Crowding	10	3.883	1.322	-0.595 710 301	3 s, 150 ms
SteadyState	100	5.614	1.424	-0.595 691 672	16 s, 627 ms
Incremental	200	4.291	1.632	-0.595 638 302	4 s, 900 ms
Incremental	100	5.163	1.045	-0.595 610 412	2 s, 817 ms
SteadyState	10	4.286	1.821	-0.595 585 401	1 s, 975 ms
Incremental	10	3.989	1.792	-0.595 532 833	925 ms
Incremental	150	6.267	1.126	-0.595 483 555	3 s, 763 ms
Incremental	50	4.368	2.124	-0.595 288 950	1 s, 602 ms
BOBYQA	N/A	0.9330	2.157	-0.586 794 255	1 s, 654 ms

**Table 3.2.3.:** The performance of the genetic algorithms with the multiple minimum surface of the 95 term  $H_2^+$  wavefunction. The optimisation algorithms are also included for comparison.



### 3. Computer Programming

designed for the purpose of employing Genetic Algorithms. This was not the only possible choice for a GA library available as other libraries including `dlib` have Genetic Algorithms. The `GAlib` has the advantage of usability and a well documented and modular design. The code was written by Dr Matthew Wall at Massachusetts Institute of Technology for his PhD. Upon completion of his PhD the code was released under the MIT license (see Appendix F.10).

#### Processing the Results

After calling the optimisation routine the calculations are now complete and the next step of the program is to process the results. If the eigenvector was requested it is saved into a file as specified by the command-line option `-sv FILENAME`. The program calculates the expectation values  $\langle \hat{T} \rangle$ ,  $\langle \hat{V} \rangle$  and  $\langle \eta \rangle$  (virial condition see Section 4.4.3 on page 124) routinely and prints them along with the optimised NLPs and the energy. The final section of the `3Body` program simply calls the appropriate destruction operators on the classes to deallocate the memory and clean up the program. The following is a simple example of this process:

#### Processing the Results

```
1 cout << "Energy=" << DtoStr(EigenResults.DesiredEigen().Energy) <<
  ↳ "\n"; //DtoStr is a helper function that prepares the number for
  ↳ printing by converting it to a string with the appropriate number
  ↳ of significant figures displayed. DesiredEigen is the requested
  ↳ root specified previously with TheDiagonaliser->SetRoot(1);
2 bool CalcedKE = false; bool CalcedPE = false;
3 T KE; T PE;
4 TheDiagonaliser-
  ↳ >CalculateT_V(EigenResults,TheDiagonaliser->GetRoot());
  ↳ //Calculate the expectation values  $\langle T \rangle$  and  $\langle V \rangle$ . Where
  ↳ TheDiagonaliser->GetRoot() is the requested root specified
  ↳ previously with TheDiagonaliser->SetRoot(1);
5 //  $\langle T \rangle$ 
```

### 3. Computer Programming

```
6 pair<bool,long> SearchResult =  
  ↳ EigenResults.DesiredEigen().ExpectationValues.HasExpectation  
  ↳ (" $\langle T \rangle$ "); //Look to see if the expectation was calculated  
  ↳ successfully by the EigenSolver class. If it has then  
  ↳ SearchResult.first is set to true and SearchResult.second is set  
  ↳ to the index to that expectation value  
7 if (SearchResult.first==true) {  
8   KE = EigenResults.DesiredEigen().ExpectationValues[SearchResult  
  ↳ .second].Value; //Extract the expectation value  
9   cout << " $\langle T \rangle$ =" << DtoStr(KE) << "\n"; CalcedKE = true; //print the  
  ↳ result  
10 }  
11 //⟨V⟩  
12 SearchResult =  
  ↳ EigenResults.DesiredEigen().ExpectationValues.HasExpectation  
  ↳ (" $\langle V \rangle$ ");  
13 if (SearchResult.first==true) {  
14   PE = EigenResults.DesiredEigen().ExpectationValues[SearchResult  
  ↳ .second].Value;  
15   cout << " $\langle V \rangle$ =" << DtoStr(PE) << "\n"; CalcedPE = true;  
16 }  
17 //Virial ( $\eta$ )  
18 if ((CalcedKE==true) && (CalcedPE==true)) { // Both were calculated  
  ↳ successfully  
19   T Virial_2 = ((PE/KE)+2.0f); //Calculate the virial condition  
20   cout << " $\langle V \rangle / \langle T \rangle + 2$ =" << Virial_2 << "\n";  
21 }  
22 //Save eigenvector  
23 LMN *lmns = new LMN(LMN::Symmetric,TheMatrix->GetSize()); string  
  ↳ fileloc = "EigenVector.sv";
```

```

24 typename Eigen::EigenPair<T>::SaveTypes ST =
    ↪ Eigen::EigenPair<T>::SaveTypes::Maple; //Specify that we will
    ↪ save the eigenvector to a Maple readable format. This is the only
    ↪ format currently programmed but other formats can be added later
    ↪ if needed.
25 EigenResults.DesiredEigen().Save(fileloc, ST, Vars, *lmns); //This
    ↪ will save the eigenvector with its variables and requires the LMN
    ↪ class that contains the triple index scheme we use for the
    ↪ Laguerre polynomials (See Appendix A).

```

#### 3.2.2. Expectation

The second C++ program written was the **Expectation** program. This program takes as input a recursion relation file as generated by Maple and the file with the eigenvector as generated by the **3Body** program (with the option `-sv FILENAME`) (this eigenvector file will be referred to as a save vector (sv) file). The recursion relation file is the specific recursion relation for the Laguerre polynomial wavefunction used here to calculate certain operators. Not all operators have a recursion relation form that enable the use of series solution but when they do it is  $\approx 1000$  times faster to calculate using the series solution recursion relation in this program than it is to calculate the expectation value in the Maple program using the `int` command. The operators that can be calculated using this **Expectation** program and used in this work includes:

- $\langle T \rangle$
- $\langle V \rangle$
- $\langle \eta \rangle$
- $\langle r_i^n \rangle$ , where  $n = -1, 0 \dots 3$ ,  $i = 1, 2, 3$
- $\langle \delta(r_i) \rangle$ , where  $i = 1, 2, 3$

This C++ program is relatively simple, the majority of the work is done by the **Eigen** library and unlike **3Body** does not need to call the **EigenSolver** or **ParameterSearcher**

### 3. Computer Programming

classes. The **Expectation** program performs the following steps:

- Read the eigenvector ( $u_i$ ) and variables from the sv file.
- Create the matrices using series solution ( $M$  for the expectation value matrix and  $S$  for the overlap matrix) using the recursion relations generated by Maple.
- Perform the linear algebra operation  $\frac{u_i.M.u_i}{u_i.S.u_i}$  to give the expectation value.
- Process the results

The **Expectation** program is run from the command line with the following arguments. Where “|” denotes “or” for arguments which need to be one of a specific set.

#### Expectation program's arguments

```
1 Expectation -r RecursionRelationFile.rr -sv SaveVetorFileFromCPP.sv
  ↪ -sym|-asym|-antisym
2
```

### Read the Eigenvector

To read the eigenvector as generated by the **3Body** program is incredibly simple from the point of view of the **Expectation** program. It is done with a few lines of code. The first step is to create the object to hold the eigenvalue and vector, the next step creates another object to save the variables contained in the sv file. Then a call to load the sv file is made.

#### Reading the SV File

```
1 string SaveVector = "SaveVetorFileFromCPP.sv"
2 EigenPair<T> EigPair(0); //Create object to hold eigenvalue and
  ↪ vector
3 VariableSet<string> Vars; //Create object to hold variables from the
  ↪ sv file.
4 Vars = EigPair.Load(SaveVector); //Load the sv file into the
  ↪ EigenPair and read the variables into the VariableSet.
```

#### Create the Matrices

To generate the matrices in the `Expectation` program the same `Matrix` class used in the `3Body` program is used. As both programs link to the shared library `Eigen` the code for the `Matrix` class is not repeated but reused. Two matrices are created from the recursion relations the  $M$  matrix and  $S$  matrix as `Matrix_RR_JIT` classes. The recursion relation has markers (see Section 3.3.1) to identify which matrix each term in the recursion relation belongs. Markers are used to separate the left and right side of the equation  $\langle \psi | \hat{A} | \psi \rangle = a \langle \psi | \hat{S} | \psi \rangle$  where  $\hat{A}$  is an appropriate operator and  $\hat{S}$  is the overlap operator. In future it may be desirable to use instead of markers a more rigorous separation of these terms such as separate files for greater readability and ease of error tracing.

#### A Simplified Example of how Expectation Loads the Matrices

```

1 //Creating the rules
2 bool KERHS=false;
3 vector<string> WithT{""};
4 vector<string> WithoutT(0);
5 vector<string> WithV{""};
6 vector<string> WithoutV(0);
7 vector<string> WithO{"ss"}; //Create a set of rules that inform the
    ↳ Matrix class that anything marked with ss is the overlap matrix
    ↳ (everything else will go into the expectation value matrix)
8 vector<string> WithoutO(0);
9 MatrixRules Rules =
    ↳ MatrixRules(WithT,WithoutT,WithV,WithoutV,WithO,WithoutO, KERHS);
10 LMN::SymmType Symm = LMN::SymmType::Symmetric;
11 Matrix_RR_JIT<T> M(FileName,(int)RR_MatSize, Symm, Rules, Vars);
    ↳ //Create the Overlap Matrix
12 ContiguousMemoryMatrix<T> ExpMatrix = M.GetMatrix(MatrixPart::HH);
    ↳ //Create the numerical matrix from the algebraic matrix and save
    ↳ it to a ContiguousMemoryMatrix class that stores the numbers in a
    ↳ single array in row then column order.
13 ContiguousMemoryMatrix<T> OvMatrix = M.GetMatrix(MatrixPart::SS);

```

### Calculating the Expectation Value

The expectation value is then calculated using the following block of code that performs the inner product of the eigenvector with the matrix and then the eigenvector again. The expectation value is then ready to be printed to the terminal.

```

Calculating the eigenvalue from the Matrix and eigenvector

1 long RR_MatSize = M.GetSize();
2 T ExpValue = 0;
3 T OvValue = 0;
4 for (long x=0;x<RR_MatSize;x++)
5 {
6     for (long y=0;y<RR_MatSize;y++)
7     {
8         ExpValue+=Vec[x]*ExpMatrix[x][y]*Vec[y];
9         OvValue+=Vec[x]*OvMatrix[x][y]*Vec[y];
10    }
11 }
12 cout << "Expectation Value=" << DtoStr(ExpValue) << "\n";
13 cout << "Overlap Value=" << DtoStr(OvValue) << "\n";
14 cout << "<Exp>/<Ov>=" << DtoStr(ExpValue/OvValue) << "\n";

```

### 3.2.3. HF2e-Integrals

The HF two electron integrals in Chapter 7 were calculated using the C++ program HF2e-Integrals. Compared to the other C++ programs discussed in this chapter it is relatively simple. It calculates the HF two electron integrals analytically and saves them to file using the maths described in Section 7.3.2 on page 215.

The core of this C++ code calculates the integrals in a quadruple **for** loop. These **for** loops iterate over all combinations of  $p$ ,  $q$ ,  $r$  and  $s$  and if the unique integral for that combination hasn't been calculated it will calculate it and save it to the array UniqueGTerms.

### 3. Computer Programming

The core for loop of HF2e-Integrals

```
1 long numberOfTerms=20; //This is set from the command line arguments,
   ↳ but we set it to 20 in this example
2 MODES MODE = MODES::CPU; //This is set from the command line
   ↳ arguments, but we set it to CPU in this example
3 PosIndex=0:
4 for (long p=1; p<=numberOfTerms; p++) { //We loop over all
   ↳ combinations of p,q,r,s
5     for (long q=1; q<=p; q++) {
6         for (long r=1; r<=numberOfTerms; r++) {
7             for (long s=1; s<=r; s++) {
8                 long QuadIndex = QuadrupleIndex(p, q, r, s); //This
   ↳ calculate a unique index from the quadruple index by
   ↳ applying the equations detailed in Appendix A
9                 if (HasIndex(UniqueGTerms, NumberofUniqueGTerms,
   ↳ QuadIndex)==false) //We check that this hasn't already
   ↳ been calculated
10                {
11                    UniqueGTerms[PosIndex].QIndex.Index=QuadIndex;
12                    UniqueGTerms[PosIndex].QIndex.p=p;
   ↳ UniqueGTerms[PosIndex].QIndex.q=q;
13                    UniqueGTerms[PosIndex].QIndex.r=r;
   ↳ UniqueGTerms[PosIndex].QIndex.s=s;
14                    //The main work is done by CalcIntegral which uses the
   ↳ analytical integral discussed in Section 7.3.2 on
   ↳ page 215
15                    UniqueGTerms[PosIndex].value=CalcIntegral(p-1, q-1, r-1,
   ↳ s-1,MODE); //zero based index so -1
16                    PosIndex++;
17                    cout << "Calculated " << PosIndex << " of " << posMax << "
   ↳ (" << round(PosIndex*100/posMax) << "%)\r";
18                    flush(cout);
19                }
```

### 3. Computer Programming

```

20     }
21 }
22 }

```

The function `CalcIntegral` calculates the HF two electron integral with the index  $p$ ,  $q$ ,  $r$  and  $s$ . There are two modes in which this is calculated **RAM** and **CPU**. As discussed in Section 7.3.2 on page 215 the **RAM** mode was used to calculate these integrals however in this discussion we will deal only with the **CPU** method as it is conceptually simpler. Mathematically the **RAM** and **CPU** methods are identical but the method of implementation greatly effects the speed of calculation. The following piece of code is a simplified example of the **CPU** method of calculation.

The `CalcIntegral` function using **CPU** intensive mode

```

1 mpreal CalcIntegral_CPU(long p,long q,long r,long s)
2 {
3     mpreal Result=0; mpreal Pi=atan(mpreal(1))*4;mpreal Pi2=pow(Pi, 2);
4     mpreal LinOp1;mpreal LinOp2;mpreal twopow; //Declared once and
    ↪ reused to avoid reinitialising the computationally expensive
    ↪ mpreal type
5     //For full details on the maths in this code See Eq. (7.3.29) on
    ↪ page 221
6     //These for loops iterates over the sums  $\sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v \sum_{a_i}^{p_i+q_i} \sum_{b_i}^{u_i+v_i}$ 
7     for (long pi=0; pi<=p; pi++) {
8         for (long qi=0; qi<=q; qi++) {
9             for (long ui=0; ui<=u; ui++) {
10                for (long vi=0; vi<=v; vi++) {
11                    for (long ai=0; ai<=(pi+qi); ai++) {
12                        for (long bi=0; bi<=(ui+vi); bi++) {
13                            long tvar = ai+bi-pi-qi-ui-vi+1;
14                            //Some calculations as they are repeated many times are
                            ↪ precalculated including  $2^n$  the result is accessed from the
                            ↪ array twopowtable
15                            twopow=twopowtable[abs(tvar)];

```



### 3. Computer Programming

```

16     if (tvar<0){
17         twopow=1/twopow;
18     }
19     //The  $(-1)^n$  is precalculated and saved to negonepowtable
20     LinOp1 =negonepowtable[(pi+qi+ui+vi+1)]*twopow*Pi2;
21     //  $a_i^2 - 2a_i b_i + b_i^2 - p^2 - 2p_i q_i - 2p_i u_i - 2p_i v_i - q_i^2 - 2q_i u_i - 2q_i v_i - u^2 -$ 
22      $\hookrightarrow 2u_i v_i - v^2 + a_i + b_i - 7p_i - 7q_i - 7u_i - 7v_i - 10$ 
23     LinOp2 =  sq(ai) -2*ai*bi +sq(bi) -sq(pi) -2*pi*qi -2*pi*ui
24      $\hookrightarrow -2*pi*vi -sq(qi) -2*qi*ui -2*qi*vi -sq(ui) -2*ui*vi$ 
25      $\hookrightarrow -sq(vi) +ai +bi -7*pi -7*qi -7*ui -7*vi -10;$ 
26     LinOp1*=LinOp2;
27     //(pi + qi + ui + vi - ai - bi)! (pi + qi)! (ui + vi)! p! q! u! v!
28     //Factorials are precalculated and saved to factab
29     LinOp1*=factab[(pi+qi -ai+ui+vi -bi)] *factab[(pi+qi)]
30      $\hookrightarrow$  *factab[(ui+vi)] *factab[(p)] *factab[(q)]
31      $\hookrightarrow$  *factab[(factab)] *factab[(v)];
32     //ui!2 vi!2 ui!2 vi!2 (pi + qi - ai)! (qi + ui - bi)! (p - pi)! (q - qi)!
33      $\hookrightarrow (u - u_i)! (v - v_i)!$ 
34     //(n!)2 is calculated with sqfac function which uses the saved
35      $\hookrightarrow$  factorials in factab
36     LinOp2=sqfac(pi)*sqfac(qi)*sqfac(ui)*sqfac(vi) *factab[(pi+qi
37      $\hookrightarrow$  -ai)] *factab[(ui+vi -bi)] *factab[(p -pi)] *factab[(q
38      $\hookrightarrow$  -qi)] *factab[(factab -ui)] *factab[(v -vi)];
39     LinOp1/=LinOp2;
40     Result+=LinOp1;
41 }
42 }
43 }
44 }
45 }
46 }
47 }
48 }
49 }
50 }
51 }
52 }
53 }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
61 }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }
88 }
89 }
90 }
91 }
92 }
93 }
94 }
95 }
96 }
97 }
98 }
99 }
100 }

```

### 3. Computer Programming

The code is relatively simple and performs the required arithmetic operations. The code also has parallel implementation but this has been left out to enhance readability.

#### 3.2.4. Eigen

The **Eigen** library contains all the relevant pieces of code to run the three-body calculations in C++. The code contains over 150 classes and over 70 000 lines of code. A general idea of the usage and design of the code has been discussed in Sections 3.2.1 and 3.2.2 it will not be explicitly discussed here.

The **Eigen** library is where all of the calculations actually happen, it contains a library of code which is generic to the eigenvalue problem. The **Eigen** library has all the capabilities to load a series solution recursion relation, create matrices, diagonalise them and optimise NLPs but has no idea of how it should do this, what it should do first, and how it should organise the results. The **3Body** program prepares the library, tells it what it should load and in what order, and provides the relevant instructions for output.

**External Library Boost:** Throughout the library **Eigen** in addition to those mentioned already the **Boost** library collection (version 1.58.0) is used. The **Boost** project is a collection of about 100 high quality peer reviewed libraries. The libraries are designed to be of high quality to speed development of programs, reduce bugs and “reduce reinvention-of-the-wheel”. 10 boost libraries have been accepted into the TR1 C++ standards technical report<sup>[50]</sup> and are included in the C++11 standard and most libraries have been proposed for inclusion into the TR2 report, and thus are in line for standardisation into the C++17 standard. We use the following **Boost** libraries all of which are under the Boost Software Licence (See Appendix F.8)

**Algorithm** Various functions in the library **Algorithm** were used to perform string searches and manipulations such as `boost::algorithm::to_lower`, `boost::algorithm::trim` and `boost::algorithm::starts_with`.

**Spirit** **Spirit** is a OO recursive parser where it is possible to write grammars and format descriptions for output. In this work it was used to parse text to numerical

values.

## 3.3. Maple Code

The Maple codes are not as specialised or as fast as the **3Body** program but they are much more flexible and capable of performing more general mathematics without significant programming effort. That is not to say that the code achieved here in Maple could not be achieved elsewhere for example we could have used the C++ **SymbolicC++** library<sup>[51]</sup> or the more interactive iPython<sup>[52]</sup> combined with **SymPy**.<sup>[53]</sup> We use the Maple software because it is effective for our needs and works with minimal setup. Unlike the other solutions however it is not freely available.

There is one primary piece of Maple code written for this work that generates all the recursion relations and saves them to file for use by the C++ programs discussed in Sections 3.2.1 and 3.2.2. In this section we will discuss the steps used to calculate and save the recursion relations for the Hamiltonian only as the process for all other recursion relations such as  $\langle r_i^n \rangle$  are identical.

The other Maple codes employed in this work calculate operators that could not be transformed into the Laguerre polynomial recursion relations and processed in C++ with the programs in Sections 3.2.1 and 3.2.2. In these cases the calculations were done in Maple but it is worth noting that future work could be devoted to programming these integrals in C++ where they would be faster as for example was done with the two electron HF integrals (See Section 7.3.2 on page 215 and Section 3.2.3).

### 3.3.1. Maple Recursion Relations

The C++ programs in Sections 3.2.1 and 3.2.2 use the series solution recursion relations generated by Maple to form the matrix elements in C++. The Maple program calculates the recursions relations for the operators from the Laguerre polynomial recursion relations. The details of these relations are given in Section 4.2.5 on page 96. The Maple program has the following stages:

- Define the operators in triangular  $r_i$  coordinates.

### 3. Computer Programming

- Convert the operators to perimetric and then scaled perimetric form (See Section 4.2.1 on page 89).
- Apply the operator to the general form of the wavefunction basis  $e^{-\frac{1}{2}(u+v+w)} L_l(u)L_m(v)L_n(w)$ .
- Apply the Laguerre polynomial recursion relations to eliminate derivatives and powers of the coordinates.
- Save the 57 term recursion relation to a file the C++ programs can read.

These steps are now briefly explained in terms of the Hamiltonian operator but these steps can apply to any operator that satisfies the Laguerre polynomial recursion relations.

#### Define the Operator and Convert to Scaled Perimetric Coordinates

The Hamiltonian operator is defined in the following way. Both finite mass and the like particle interaction are included ( $r_3$ ).

##### Assigning the Hamiltonian Operators

```

1 T:=psi->-h^2/2*(Sum(1/(mu[i]*r[i]^2)*Diff(r[i]^2*Diff(psi,r[i]),
  ↪ r[i])+Sum(b[i,j]/(mu[i,j])*Diff(psi,r[i],r[j]),j=(i+1)..3),
  ↪ i=1..3)):
2 V:=psi-> e^2/(4*Pi*epsilon[0])*Sum(Sum(Z[i]*Z[j]/r[i,j],
  ↪ j=(i+1)..3),i=1..3) *psi:
3 S:=psi->E*psi:
4 mu[1] := 1/(1/m[1] + 1/m[3]):
5 mu[2] := 1/(1/m[2] + 1/m[3]):
6 mu[3] := 1/(1/m[1] + 1/m[2]):
7
8 mu[1,2] := m[3]:
9 mu[1,3] := m[1]:
10 mu[2,3] := m[2]:
11
12 b[1,2]:=(r[1]^2+r[2]^2-r[3]^2)/(r[1]*r[2]):
13 b[1,3]:=(r[1]^2+r[3]^2-r[2]^2)/(r[1]*r[3]):

```

### 3. Computer Programming

```
14 b[2,3]:=(r[2]^2+r[3]^2-r[1]^2)/(r[2]*r[3]):
```

The coordinate transformation is applied with the Maple package PDETools, that perform coordinate transformations on partial differential equations given a set of coordinate transformations and their reverse transformation. These forward transformation from  $r_i$  to  $z_i$  are stored in  $R_i$  and reverse in  $rR_i$ . The Jacobian is also calculated using the determinant of the Jacobi Matrix.

#### Transforming the Coordinates to Perimetric

```
1 R1:=(z[2]+z[3])/2:#Forward transformation
2 R2:=(z[1]+z[3])/2:
3 R3:=(z[1]+z[2])/2:
4 tr:={r[1]=R1,r[2]=R2,r[3]=R3}:
5 rR1:=-r[1]+r[2]+r[3]:#Reverse transformation
6 rR2:=r[1]-r[2]+r[3]:
7 rR3:=r[1]+r[2]-r[3]:
8 rtr:={z[1]=rR1,z[2]=rR2,z[3]=rR3}:
9 nv:={z[1],z[2],z[3]}:#New variable names
10
11 Jac:=Student['MultivariateCalculus']['Jacobian']([R1,R2,R3],[z[1],
    ↪ z[2],z[3]],output=determinant);
12 g:=r[1]*r[2]*r[3]*8*Pi^2: #This is the volume element of Cartesian
    ↪ to triangular coordinate change.
13 KE:=psi->T(psi):
14 PE:=psi->V(psi):
15 OV:=psi->S(psi):
16
17 wavefn:=psi(u,v,w): #The wavefunction is left undefined for now
18 KE:=expand(simplify(PDEtools[dchange](tr,(KE(wavefn)*g*Jac),nv,rtr)
    ↪ )):
19 PE:=expand(simplify(PDEtools[dchange](tr,(PE(wavefn)*g*Jac),nv,rtr)
    ↪ )):
```

### 3. Computer Programming

```
20 OV:=expand(simplify(PDEtools[dchange](tr,(OV(wavefn)*g*Jac),nv,rtr)
    ↪ ))):
```

The same procedure is then applied to the coordinate transformation from  $z_i$  to  $u, v, w$ . Except PDETools now also requires the non-linear variational parameters  $A, B, C$  but the reverse transformation is not required because the transformation is simple.

#### Transforming the Coordinates to Scaled Perimetric

```
1 U:=A*z[1]:
2 V:=B*z[2]:
3 W:=C*z[3]:
4 NLP:={A,B,C}:
5 tr:={z[1]=solve(u=U,z[1]), z[2]=solve(v=V,z[2]),
    ↪ z[3]=solve(w=W,z[3])}:
6 nv:={u,v,w}:
7
8 Jac:=Student['MultivariateCalculus']['Jacobian']([solve(u=U,z[1]),
    ↪ solve(v=V,z[2]),solve(w=W,z[3])],[u,v,w],output=determinant);
9
10 KE:=eval(expand(simplify(PDEtools[dchange](tr,KE*Jac,nv,params=NLP)
    ↪ ))) :
11 PE:=eval(expand(simplify(PDEtools[dchange](tr,PE*Jac,nv,params=NLP)
    ↪ ))) :
12 OV:=eval(expand(simplify(PDEtools[dchange](tr,OV*Jac,nv,params=NLP)
    ↪ ))) :
```

#### Apply the Wavefunction and Recursion Relations

The next step informs the Maple program about the form of the wavefunction and allows Maple to evaluate the differential terms with the non-linear variational parameters in the wavefunction. As the Laguerre polynomials do not need to be dealt with yet they are left in the form  $F_{l,m,n}(u,v,w) = L_l(u)L_m(v)L_n(w)$ .

### 3. Computer Programming

#### Applying the Exponential Term

```
1 psi(u,v,w):=exp(-u/2-v/2-w/2)*F[1,m,n](u,v,w):
2 KE:=expand(simplify(calculate(KE))):
3 PE:=expand(simplify(calculate(PE))):
4 OV:=expand(simplify(calculate(OV))):
```

The appropriate markers for the C++ programs are then applied these are used to extract the operators in the C++ programs from a single recursion relation expression. These have no mathematical significance but are used to find terms for each operator and work like a filter. A single term can have more than one marker. The markers are as follows

*hh* For the terms that form the left hand side of the general eigenproblem.

*ss* For the terms that form the right hand side (overlap) of the general eigenproblem.

*t* For the terms that constitute the kinetic energy of the Hamiltonian.

*p* For the terms that constitute the potential energy of the Hamiltonian.

*o* For the terms that constitute the overlap of the Hamiltonian.

We apply these markers and then generate the recursion relations using a specially written function (`ApplyRecursionRelations`) that is stored in the file `RecursionRelations.mpl`.

#### Applying The Recursion Relations

```
1 Expr:=KE*t*hh+PE*p*hh+OV*o*ss:
2 Expr:=expand(simplify(-Expr)): #Our overlap matrix is negative
   ↪ definite we need positive definite for Cholesky Decomposition
   ↪ therefore multiple through by -1.
3
4 read "RecursionRelations.mpl":
5 RR:=ApplyRecursionRelations(Expr):
```

### 3. Computer Programming

**RecursionRelations.mpl** The code written in file RecursionRelations.mpl has the function `ApplyRecursionRelations` this is a fairly useful generic function to apply recursion relations to functions with Laguerre polynomial terms. Or more specifically functions with the  $F_{l,m,n}(u, v, w)$  terms. It works by looping over every term searching for differentials of  $F$  with respects to  $u$ ,  $v$  or  $w$ , and for powers of  $u$ ,  $v$  or  $w$ . The code to perform this is quite complex and so shall not be detailed here. The output is an expression with no differentials or powers of  $u$ ,  $v$  or  $w$ . The function takes as input whole expressions containing Laguerre polynomial terms such as  $\frac{\partial F_{l,m,n}(u,v,w)}{\partial u}$  and reduces them down to simple functions of the form  $F_{l+dl,m+dm,n+dn}$  without any differentials or powers of  $u$ ,  $v$  or  $w$  where  $dl$ ,  $dm$  and  $dn$  are integers (for the Hamiltonian operator these are in the set  $-2, -1, 0, 1, 2$ ).

#### Save The Recursion Relations to File

These relations are then saved to file. The format of the file is fairly simple. It is in text format with the following repeating format: `| [dl,dm,dn],RecursionRelation`. Before saving the relation optionally a transformation to convert the wavefunctions to either the  $A$ ,  $AC$ ,  $ABC$ ,  $K$  or the Pekeris wavefunction is applied (See Section 4.2.8 on page 105).

#### Maple code to Save the Recursion Relations

```
1 SaveRRToFile:=proc(RR,FileName)
2   local Fterms, size, index, dl, dm, dn, Expr, fd:
3   Fterms := (indets(RR,function)):
4   size:=nops(Fterms):
5   fd:=fopen(FileName, WRITE, TEXT):
6   for index from 1 to size do
7     dl:=eval(op([0,1],Fterms[index]),[l=0,m=0,n=0]):
8     dm:=eval(op([0,2],Fterms[index]),[l=0,m=0,n=0]):
9     dn:=eval(op([0,3],Fterms[index]),[l=0,m=0,n=0]):
10    Expr:=coeff(RR,Fterms[index]):
11    fprintf(fd,"| [%A,%A,%A],%A",dl,dm,dn,Expr):
12  end do:
13  fclose(fd):
```



### 3. Computer Programming

```
14 end proc:
15
16 #ABC wavefunction
17 Transformation:=[]:
18 SaveRRToFile(expand(simplify(eval(RR,Transformation))), "ABC/RR.rr"):
19 #AC wavefunction
20 Transformation:=[B=A]:
21 SaveRRToFile(expand(simplify(eval(RR,Transformation))), "AC/RR.rr"):
22 #A wavefunction
23 Transformation:=[B=A, C=2*A]:
24 SaveRRToFile(expand(simplify(eval(RR,Transformation))), "A/RR.rr"):
25 #K wavefunction
26 Transformation:=[A=1, B=1, C=2, t=t*ss/hh, E=K*ep]:
27 SaveRRToFile(expand(simplify(eval(RR,Transformation))), "K/RR.rr"):
28 #PEKERIS wavefunction
29 Transformation:=[A=1, B=1, C=2, t=t*ss/hh, E=ep]:
30 SaveRRToFile(expand(simplify(eval(RR,Transformation))),
  ↪ "PEKERIS/RR.rr"):
```

This code is used to calculate the recursion relations for all the operators used in the C++ programs in Sections 3.2.1 and 3.2.2. These operators include  $\hat{T}$ ,  $\hat{V}$ ,  $\hat{r}_i$ ,  $\hat{r}_i^2$ ,  $\hat{r}_i^3$ ,  $\hat{r}_i^{-1}$ ,  $\delta(\hat{r}_i)$ ,  $\hat{\tau}_i$  and  $\hat{f}$ . The program is not as efficient as it could be particularly with the function `ApplyRecursionRelations` which is particularly slow with the operators  $\hat{r}_i^3$  because of the recursion involved and the number of terms. However this is not an issue as once a recursion relation is calculated for a particular operator it need not be calculated again and may be reused repeatedly for any particular type of wavefunction that has been implemented in this work.

#### 3.3.2. Other Maple Programs

The other Maple programs in this work calculate the operators that cannot be fully reduced to a recursion relation and or used in series solution. These include  $\langle \delta(r_i - r) \rangle$  (where  $r \neq 0$ ), because with this expectation value the integrals do not range over 0 to  $\infty$  stopping the series solution from functioning correctly, and the

### 3. Computer Programming

cusps as calculated by  $\frac{\langle \frac{\partial \psi}{\partial r_i} \delta(r_i) \rangle}{\langle \delta r_i \rangle}$ , because the expression contains  $r_i^{-1}$  terms that can not be reduced by the Laguerre polynomial's recursion relations.

Each of these programs is unique but follows a general design philosophy. First the wavefunction is loaded into Maple from the C++ sv file. Any coordinate transformation is applied and the Jacobian of the transformation is assigned. The expectation value is then calculated using Maple's built in **int** command. The results are then printed. If the expectation value produces a range of values such as that by  $\langle \delta(r_i - r) \rangle$  when  $r$  is varied the values are saved to separate file named with the same name as the sv file but with a small description appended to the name. For example **\_dr3** is appended when calculating the expectation function  $\langle \delta(r_3 - r) \rangle$ .

The Maple GUI is not used for these Maple calculations because the Maple GUI does not keep a record of the calculation results. Instead the command-line interface for the Maple program is used and the output is appended to the log files from the 3Body programs to keep all results for a particular calculation together using the command: `maple -q MAPLESCRIPT >> LOGFILE`.

#### Calculating the Cusps-Loading The Wavefunction

The following Maple code will calculate the cusps of a system. This is a good example of the Maple codes that just calculate a single value.

##### The Calculation of the Cusps in Maple

```
1 Digits:=32: #Use 32 digits of precision
2 L:=LaguerreL:
3 u:=z1*A:v:=z2*B:w:=z3*C: #Convert scaled perimetric coordinates to
   ↪ perimetric coordinates
4 z1:=r2+r3-r1:z2:=r1+r3-r2:z3:=r1+r2-r3: #Convert perimetric
   ↪ coordinates to internal triangular coordinates
5
6 filename:="SAVEVECTOR": #The name of the sv file is set here
7 read filename: #The sv file is loaded. The variable 'F' now contains
   ↪ the radial wavefunction.
8 #Detect the wavefunction type
9 if (C = 'C') #If C is undefined then set C to A+B
```

### 3. Computer Programming

```

10 C:=A+B: #AB psi
11 fi:
12 if (B = 'B') then #If B is undefined then set B to A
13   B:=A: #A
14 fi:
15 if (A = 'A') then #If A is undefined then we are using the PEKERIS
    ↪ style wavefunctions
16   A:=ep: #PEKERIS and K
17 fi:

```

The piece of code above is identical in all Maple codes for calculating expectation values in this work and so will not be repeated again, in later examples. It loads the wavefunction from the sv file and by doing so the variable F is assigned to the radial part of the wavefunction as calculated by the C++ program 3Body. This radial part has the general form:  $\sum_{l,m,n} C_{l,m,n} L_l(u) L_m(v) L_n(w)$ .

This next part of the Maple code is unique to the expectation value being calculated. Here the Jacobian is assigned and the expectation values integrated using Maple's built in **int** routine.

#### Calculating the Cusps-The Integrals

```

1 g:=4*Pi*r3*r2: #The Jacobian of the transformation involved in this
    ↪ expectation value
2
3 psi:=expand(simplify(exp(-u/2-v/2-w/2)*(F))): #The wavefunction as a
    ↪ combination of the 'F' function as loaded from the sv file and an
    ↪ exponential function that represents the asymptote of the
    ↪ wavefunction (See Section 4.2.8 on page 105).
4
5 dpsi:=simplify(diff(psi,r1)): #This wavefunction is differentiated
6 v31:=int(eval(simplify(eval(psi*dpsi*g,[r2=r3])),r1=0),
    ↪ r3=0..infinity): #The expectation value of  $\left\langle \frac{\partial \psi}{\partial r_1} \delta(r_1) \right\rangle$ 

```

### 3. Computer Programming

```
7 dr1:=int(eval(simplify(eval(psi*psi*g,[r2=r3])),r1=0),
  ↪ r3=0..infinity): #This is the expectation value of the Dirac
  ↪ delta <math>\langle \delta(r_1) \rangle</math>
8
9 Nuclear_cusp:= v31/dr1:
10
11 printf("v31 Numerator: %A\n",v31):
12 printf("v31 Denominator: %A\n",dr1):
13 printf("v31 cusp value Scaled: %A\n",Nuclear_cusp):
```

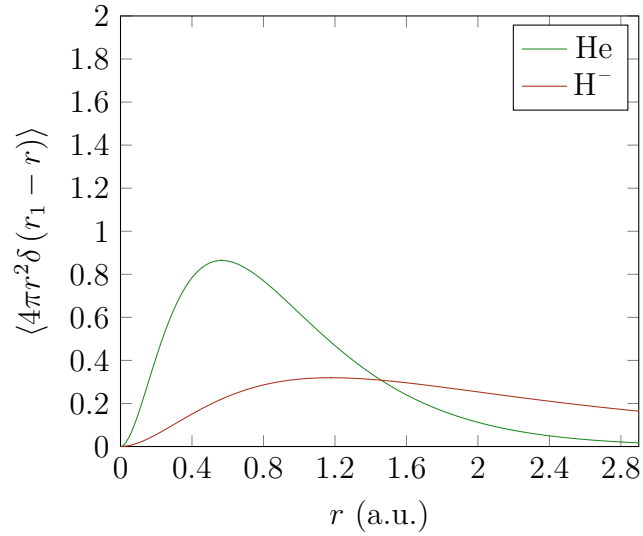
The Maple scripts have this general form to calculate the two cusp values  $\nu_{31}$  and  $\nu_{21}$ . For the particle densities however a more complex code exists.

#### Calculating Particle Densities

In this work when calculating particle densities and radial distribution functions such as  $\langle \delta(r_i - r) \rangle$  and  $\langle 4\pi r^2 \delta(r_{in} - r) \rangle$  the Maple code grows in complexity. Each piece of code generates a series of data points that are later used to form plots such as that in Figure 3.3.1. These pieces of code loop over a series of values and save them to a particle density (pd) file with the arbitrary extension .pd. The code also performs some additional checks and calculation to ensure we obtain decent plots. First it will check that the values have reduced to  $\approx 100^{\text{th}}$  of their maximum height and if not the Maple code will calculate more points until the density reduces to  $100^{\text{th}}$  of its maximum height. The Maple code will then locate the maximum value precisely using Maple's built in quadratic optimisation routine. While locating the maximum it will log the steps it takes to find the maximum to the pd file this ensures that the peaks on the plot have more points describing them, this will help any sharp peaks to look smoother. In some cases the command **fsolve** is used to also identify the full width half maximum of the peaks but this was not used in the work presented in this thesis and so shall not be discussed.

In this first block of code a special function is defined to help perform the integration named **aint**. This function will first attempt full analytical integration but when this fails or numerical instabilities are identified this function will use approximate

### 3. Computer Programming



**Figure 3.3.1.:** An example particle density plot.

integration with Riemann sums with 100 000 partitions. This number of partitions maintains an accuracy of at least 9 s.f. but usually more.

Calculation of  $\langle \delta(r_3 - r) \rangle$ -The Integration

```

1 partitions:=100000:
2 aint:=proc(f,range)
3   local val, infity:
4   infity:=20:
5   try
6     val:=int(f,range): #try full integration first in almost all
        ↳ cases this succeeds
7     if type(val, 'undefined') then #If full integration fails use
        ↳ approximate ints using Riemann sums
8       val:=ApproximateInt(f,eval(range,infinity=infity),
        ↳ partition=partitions):
9     elif type(val, 'numeric') then
10      if (abs(val/Z3)>1000) then #If the number is very large then
        ↳ we have numerical instability and we recalculate with
        ↳ approximate integration with Riemann sums
11      val:=ApproximateInt(f,eval(range,infinity=infity),
        ↳ partition=partitions):
12    fi:

```

### 3. Computer Programming

```

13   fi:
14   catch: #If an error occurs then use approximate ints with Riemann
        ↪ sums
15   val:=ApproximateInt(f,eval(range,infinity=infty),
        ↪ partition=partitions):
16   end try:
17   return val:
18 end proc:
19
20 gDirac:=4*Pi*r1*r2/(2*r3):
21 #Jacobian on u->ri conversion
22 gDirac:=gDirac/(2*C*A):
23
24 psi:=expand(simplify(exp(-u/2-v/2-w/2)*(F))):
25
26 r1:=(v/B+w/C)/2:r2:=(u/A+w/C)/2:r3:=(v/B+u/A)/2:
27
28 drval:=(r)->aInt(int(eval(psi*psi*gDirac, v=2*B*(r-(u/(2*A))))),
        ↪ w=0..infinity), u=0..2*A*r):

```

The next block of code integrates  $r$  (in  $\langle \delta(r_3 - r) \rangle$ ) over a series of values between 0 to 15 scaled by the charge. We scale by the charge because  $\langle \delta(r_3 - r) \rangle$  scales with charge in an atomic system<sup>[54,55]</sup> and this ensures we calculate the points efficiently. The pd file is saved in a text format tabulated into a table of  $r$  and  $\langle \delta(r_3 - r) \rangle$ .

#### Calculation of $\langle \delta(r_3 - r) \rangle$ -Tabulating the Data

```

1 #We set the pd file name based on the sv file name
2 PDFFileName:=(cat(FileTools[Basename](SVFileName), "_dr3.pd")):
3 #The range we scanned by the function depends on the charge of the
  ↪ wavefunction since as the charge tends towards infinity the plot
  ↪ moves closer to zero and therefore needs more details nearer the
  ↪ origin and smaller steps to better describe the density.
4 Min:=0.0/abs(Z3):

```

### 3. Computer Programming

```

5 Max:=15/abs(Z3):
6 Step:=0.05/abs(Z3):
7 printf("Calculating %delta(r_3-r) Particle Densities\nSaving to the
  ↪ file %A\nFrom %A to %A in steps of
  ↪ %A\n",PDFFileName,Min,Max,Step):
8 fd := fopen(PDFFileName, WRITE, TEXT):
9 fprintf(fd,"Calculated delta(r_3-r) v0.01a\n"):
10 fprintf(fd,"r\tdelta(r_3-r)\n"):
11 #Keeps track of the maximum values to help locate the maximum for
  ↪ the optimisation function later
12 maximumvalue:=-infinity: maximumr:=-infinity:
13 maximumDvalue:=-infinity: maximumDr:=-infinity:
14 for r from Min by Step to Max do
15   ExpVal:=drval(r):
16   fprintf(fd,"%A\t%A\n",r,ExpVal):
17   if (ExpVal>maximumvalue) then #This is used to locate roughly the
     ↪ maximum of  $\langle \delta(r_3 - r) \rangle$ 
18     maximumvalue:=evalf(ExpVal):
19     maximumr:=r:
20   fi:
21   if (ExpVal*4*Pi*r^2>maximumDvalue) then #This is used to locate
     ↪ roughly the maximum of  $\langle 4\pi r^2 \delta(r_3 - r) \rangle$ 
22     maximumDvalue:=ExpVal*4*Pi*r^2:
23     maximumDr:=r:
24   fi:
25 od:

```

The following code checks that after the range of values have been calculated that the last value is approximately zero, where approximately zero is considered to be 100<sup>th</sup> of the maximum height. If not it will continue tabulating data until it is.

### 3. Computer Programming

Calculation of  $\langle \delta(r_3 - r) \rangle$ -Extending the Range

```

1  #is dr ≈ zero? if not keep plotting
2  #taking approximate zero to be 100th of the maximum value rounded
   ↪ down to the nearest 1
3  ZeroApproximate:=1*10(floor(log[10](maximumvalue/100))):
4  if (ExpVal > ZeroApproximate) then #If the function is not yet
   ↪ approximately zero keeping plotting until it is
5    StepMultiplier:=10: ReachedZero:=false: LoopCounter:=1:
6    MaxExtentions:=20: #If this many extensions to the plot is
   ↪ exceeded finish regardless
7    while ReachedZero=false do
8      Min:=Max+Step:
9      Max:=Max+Step*StepMultiplier:
10     for r from Min by Step to Max do
11       ExpVal:=drval(r):
12       fprintf(fd,"%A\t%A\n",r,ExpVal):
13       FileTools[Flush](fd):
14     od:
15     if (ExpVal <= ZeroApproximate) then
16       ReachedZero:=true:
17     elif LoopCounter >= MaxExtentions then #Reached maximum number
   ↪ of retries
18       break:
19     else
20       LoopCounter:=LoopCounter+1:
21     fi:
22   od:
23 else
24   ReachedZero:=true:
25 fi:

```

The peaks of the particle density and the radial distribution function are calculated using the NLPSolve optimisation routine. While calculating these values the intermediate steps are added to the pd file. This ensures that extra data points



### 3. Computer Programming

around the maximum are included to refine the shape of the maximum in any plots made, that could otherwise appear sharp.

#### Calculation of $\langle \delta(r_3 - r) \rangle$ -Refining the Maxima

```

1 printf("Refining maximum\n"): #Now locating the maximum using the
   ↪ NLPsolve command and its quadratic optimisation routine
2 maximummin:=maximumr-Step*5: maximummax:=maximumr+Step*5: #This will
   ↪ roughly bracket the maximum based on previous calculations
3 if maximummin < 0 then
4   maximummin:=0:
5 fi:
6 Logdr:=proc(r) #This function will log the expectation value to the
   ↪ pd file and return the value the NLPsolve needs to calculate the
   ↪ maximum
7   local val:
8   val:=drval(r):
9   fprintf(fd,"%A\t%A\n",r,val):
10  return val:
11 end proc:
12 Opted:=Optimization[NLPsolve](evaln(Logdr(var)),
   ↪ var=(maximummin)..(maximummax), optimalitytolerance=3E-15, assume
   ↪ = nonnegative, maximize=true):
13 rimax:=solve(Opted[2][1],var):
14 Drimax:=Opted[1]:
15 printf("Maxima located\n"):
16 printf("delta(r_3-r) max r=%A\n",rimax):
17 printf("delta(r_3-r) max probability=%A\n",Drimax):
18
19 #A similar maximum finding routine is also employed for  $\langle 4\pi r^3 \delta(r_3 - r) \rangle$ 
20
21 printf("Calculation of delta(r_3-r) Particle Densities was completed
   ↪ in %A (s)\n", (time()-allst)):

```

The Maple code described in this section is typical of the Maple code that calculates particle densities and radial distribution functions. This style of Maple code was

used to calculate the following expectation values:  $\langle \delta(r_i - r) \rangle$ ,  $\langle 4\pi r^2 \delta(r_{in} - r) \rangle$  and  $\langle 4\pi r^2 \delta(r_{out} - r) \rangle$ .

## 3.4. Shell Scripts

The command-line programs of both C++ and Maple are employed using Shell scripts. This is because the Shell scripts can be created to produce a simple interface to these programs and can process the output efficiently. In this work the Bourne-Again shell (bash) is used in all scripts; specifically version 4.3.39(1). These scripts are designed to reduce complexity of the tasks and aid in keeping records of the calculations run. For example the main script used to run the 3Body program will automatically calculate the expectation values if it saves the sv file, it then logs the data to the system's log file. These scripts also contain the appropriate defaults; for example it will set the tolerance of the optimisation routines automatically according to the number of digits requested in the calculation. The scripts also perform more complex tasks and combine different optimisation techniques such as BOBYQA and NROpt (see Section 3.2.1) in what is referred to as the Fork Method (see Section 3.4.1).

In this section these scripts and their importance in this work is discussed.

### 3.4.1. Main Shell Scripts

These are the scripts that perform complex tasks such as the main run script that deals with the 3Body program, the Fork method and recalculation of 16 digits optimisation results with 32 digits of precision.

#### The Run Script

The main script that deals with the 3Body program is the run script. This script contains all the default settings for the 3Body program and will set the program up to use the correct non-linear variational parameters for the specified wavefunction, to use the formal asymptotic solution of the wavefunction as the initial guess of the non-linear variational parameters, and the method to optimise them. It will also perform simple sanity checks such as ensuring that  $Z_1$  and  $Z_2$  are the same sign and

### 3. Computer Programming

the  $Z_3$  is the opposite sign. In the following we briefly describe this script in terms of user input but not the more complex steps that follow this input.

The first section of the script chooses real-world variables such as the masses and charges

#### Simplified Run Script-Choosing the System

```
1 #!/bin/bash
2
3 #Set the masses of the particles. We can use the symbols for example
  ↪ p p mu and the program will correct it to, 1836.15267245,
  ↪ 1836.15267245, 206.7682843
4 m1="${e}"
5 m2="${e}"
6 m3="${infinity}"
7
8 #Set the charges
9 Z1=-1
10 Z2=-1
11 Z3=1
12
```

The next section of the script chooses the wavefunction, the number of terms and the optimisation technique.

#### Simplified Run Script-Setting the wavefunction

```
1 MATRIXSIZE="2856"
2
3 Mode="OPT"           #Select a mode. Possible modes are SP (single
  ↪ point), OPT (optimisation) and SCAN (scan).
4 MatrixType="AC"      #possible values are A AC ABC K PEKERIS
5 SYMM="SYM"          #This should be SYM ANTISYM or ASYM
6
```

### 3. Computer Programming

```
7 #This will calculate the square root of two-body energies of the
  ↳ pairs  $m_1m_3$  and  $m_2m_3$  and save them to Eh1 and Eh2 respectively.
  ↳ These are then used as the initial values of the non-linear
  ↳ variational parameters (See Section 4.2.7 on page 100).
8 source CalcFormal ${m1} ${m2} ${m3} ${Z1} ${Z2} ${Z3}
9
10 #These are the non-linear variational parameters. In opt they are
  ↳ the initial values and in scan they are the minimum values.
11 A="1*${Eh2}"
12 B="1*${Eh1}"
13 C="${A}+${B}"
14
15 #Scan only variables. If scanning also set these variables. If not
  ↳ scanning they are ignored
16 A_Max=2
17 B_Max=2
18 C_Max=4
19 A_Step=0.1
20 B_Step=0.1
21 C_Step=0.1
22
23
```

Then some calculation details such as the tolerance of the optimisation how many digits to use in the calculation and where to save the sv file are set up.

#### Simplified Run Script-Details

```
1 #This variable sets the tolerance to  $3 \times 10^{tol}$ 
2 tol=-10
3
4 sv="FILENAME.sv" #Put in name of output eigenvector file if the
  ↳ eigenvector should be saved, if blank the eigenvector file is not
  ↳ created
```

### 3. Computer Programming

```
5 arb="32" #Blank means use default precision, this is double unless a
   ↳ high tolerance is requested. If not blank it will use the given
   ↳ number of digits.
6 OptType="-bobyqa" #Possible values are blank (use the default,
   ↳ NROpt), -NROpt (From Numerical Recipes), -MacOpt (Use modified
   ↳ Numerical Recipes), -bfgs (Use Broyden-Fletcher-Goldfarb-Shanno
   ↳ algorithm), -lbfgs (Similar to -bfgs but used when there are
   ↳ hundreds of non-linear variational parameters and memory is
   ↳ limited), -cg (The dlib variant of conjugate gradient), -bobyqa
   ↳ (The BOBYQA algorithm by M.J.D. Powell excellent for optimising
   ↳ with out derivatives)
7
```

The rest of this script does not require any more input from the user. It will take these values and prepare the appropriate settings to run the **3Body** program. The design behind this script was to allow users to interact with the **3Body** program in a more friendly and informative way then just through command-line arguments. This script will also calculate the expectation values using the **Expectation** program if the sv file was requested with the **sv=** setting.

In addition to this script there exist variants for the mass Hamiltonian (See Chapter 5) and the charge Hamiltonian (See Chapter 6). These will not be discussed as they are nearly identical to this script.

#### The Fork Method

While the run script is the most complete script for running a single methodology of the **3Body** program the Fork script, which combines multiple methodologies, was the most used throughout this work. This method uses both the optimisation methods BOBYQA and NROpt to perform an initial optimisation. At the end of both calculations the program picks the better of the two methods in terms of most stable energy and then runs a final BOBYQA starting from the best non-linear variational parameters. Since two optimisation paths are followed and the algorithm takes the better of the two paths, it is called the Fork method as in fork in the road. This is

### 3. Computer Programming

not to be confused with forking a program or source repository.

This method has the advantage of sampling more of the parameter space to find the minimum values of the non-linear variational parameters. It was found that this method is capable of locating the lowest minimum of the molecular systems. The bash script itself is fairly straightforward. It starts two **3Body** programs optimising the non-linear variational parameters from their formal start value. In later versions of this script GNU parallel<sup>[56]</sup> is used to run these two jobs in parallel. The best results are then used to start a new BOBYQA optimisation to further optimise the results. By starting the BOBYQA algorithm again the trust region size will reset and this means that the algorithm will search the surrounding space for a new minimum. Another algorithm such as MacOpt would not be appropriate as they would test the gradient and decide the calculation is finished rather than search the surrounding area but BOBYQA will always actively test the surrounding parameter space since it uses no gradient information, further increasing the parameter space searched.

#### **The GA Fork Method**

This next major script did not see much use in this project since the Fork method was effective in almost all cases. This method, a combination of GA methods and the Fork method, attempts to sample as much parameter space as possible and ensure that the best non-linear variational parameters are located. This method is by far the slowest but is the most reliable at finding the optimum non-linear variational parameters. Like the Fork method the calculation starts with a parallel optimisation using BOBYQA and NROpt. After this however a GA optimisation is performed using the “Simple” algorithm (See Section 3.2.1) and a population size of 200. The non-linear variational parameters ranges provided to the GA method are  $\pm 5$  around the best results from the BOBYQA or NROpt (which ever gave the lowest energy). Once the GA optimisation completes a second parallel optimisation of BOBYQA and NROpt is performed starting from the best GA results. Finally a BOBYQA optimisation is performed with non-linear variational parameters starting from the best of these two.

This method samples a lot of parameter space and incorporates a global minimum

### 3. Computer Programming

search routine. Its purpose is to identify the best non-linear variational parameters for any system given to it. In this work it was used to test the Fork optimisation routine of  $\text{H}_2^+$  to ensure that the correct minimum was located with a 2856 term wavefunction.

Despite the complexity of this method, it is still an efficient method of locating minima compared to a scan optimisation as it locates minima in an intelligent manner with fewer steps.

#### Recalculating the Eigenvector

The final script discussed in this section is the **ReCalcVector**. This script was used every time the eigenvectors and the expectation values were calculated. The purpose of this script is to take the output of one of the above scripts in Section 3.4.1 and take their optimised values and then recalculate the eigenvector (the sv file) using more arbitrary precision. This script takes the following command-line arguments:

```
Running ReCalcVector
1 ReCalcVector --arb 32 LOGFILE
2
```

The 32 option can be changed to any digit but in this work the eigenvectors were always recalculated with 32 digits. This script was used in every recalculation of an eigenvector as it ensures that the calculation is performed always in an identical manner and that it logs the calculation to the appropriate file. In addition this script detects if the calculation uses the charge Hamiltonian, and in this case applies the appropriate maths to extract the non-linear variational parameters. This is needed because the charge Hamiltonian uses a  $Z_3$ -scaled Hamiltonian meaning that  $r_i = Z_3 r_i$ . When later using the standard Hamiltonian it is necessary to be mindful of this implicit scale on the coordinates and reapply the scale using the non-linear variational parameters. This script handles the conversion automatically and is designed to recalculate an eigenvector in a fail-safe manner. Should any other such considerations be necessary in future, this script should be adjusted to reflect these considerations.

The script also has the capability to optimise the non-linear variational parameters

### 3. Computer Programming

starting from the values in the log file but with more digits and this is achieved using the command-line option `--opt`. This was not used in the work presented in this thesis but was used for the paper written by our group entitled “Uncorrelating the correlated motion of two-electron atoms”. (In publication)

After the recalculation of the eigenvector this script recalculates the expectation values that can be quickly calculated using the recursion relations and the C++ program `Expectation`.

#### 3.4.2. Expectation Scripts

In addition to the above scripts there are a few simpler scripts that calculate the expectation values that deserve a mention. This is because they kept the work consistent and ensured that all calculations were appropriately logged. These scripts include both the C++ program (`Expectation`) and the Maple methods of calculating expectation values.

##### **CalcExpectation**

The first of these scripts to be discussed is `CalcExpectation` which calculates the expectation values using the program `Expectation`.

The script `CalcExpectation` will take as input the expectation value name that is to be calculated, the sv file, the symmetry of the wavefunction, and the optional argument of desired precision of the calculation which defaults to 16 digits if omitted.

CalcExpectation Command-line Options

```
1 CalcExpectation ExpValName SVFILE sym|antisym|asym
  ↪ PEKERIS|K|A|AB|AC|ABC [-arb <NUM>]
2
```

Where `ExpValName` is the name of the expectation value, and “|” denotes “or” where one option is required from the set of possible options and where arguments in [ ] denote optional arguments. The possible expectation value names include all expectation values that can be calculated with the C++ program `Expectation`:

**ri** Will calculate the expectation value of  $\langle r_i \rangle$  where  $i = 1, 2, 3$



### 3. Computer Programming

**ri^2** Will calculate the expectation value of  $\langle r_i^2 \rangle$  where  $i = 1, 2, 3$

**ri^3** Will calculate the expectation value of  $\langle r_i^3 \rangle$  where  $i = 1, 2, 3$

**ri^-1** Will calculate the expectation value of  $\langle r_i^{-1} \rangle$  where  $i = 1, 2, 3$

**dri** Will calculate the expectation value of  $\langle \delta(r_i) \rangle$  where  $i = 1, 2, 3$

**E** Will calculate the expectation value of  $\langle \hat{H} \rangle$

**T** Will calculate the expectation value of  $\langle \hat{T} \rangle$

**V** Will calculate the expectation value of  $\langle \hat{V} \rangle$

There are a few other expectation values not listed as they were not used in this work but in essence this scripts sets up and calculates anything the **Expectation** program calculates.

The program **CalcExpectations** takes slightly different arguments.

#### CalcExpectations Command-line Options

```
1 CalcExpectations LOGFILE sym|antisym|asym PEKERIS|K|A|AB|AC|ABC
  ↪ [-arb <NUM>]
2
```

This script will then calculate all the expectation values that the C++ program **Expectation** can calculate and saves the results to the appropriate log file. This script is used in the other scripts described in Section 3.4.1 to calculate the expectation values, ensuring that all these programs calculate the expectation values in the same manner and avoid repeated code that would have needed to be debugged individually.

#### CalcCusps

The bash script **CalcCusps** takes only the log file as input (and assumes the sv has the same name) and will calculate all the cusps using Maple. This includes both the expansion method<sup>[57]</sup> and the Dirac method<sup>[58,59]</sup> of calculating the cusps. In this work only the Dirac method results are reported as this method gave more accurate results (For calculation details see Section 3.3.2).

#### Calcdri

The script **Calcdri** is used to calculate the particle densities of our systems using the Maple code in Section 3.3.2. The script is a good example of how these scripts operate and so we shall examine it in further detail. It takes as input the following arguments, where arguments in [ ] denote optional arguments:

##### Calcdri Command-line Options

```
1 Calcdri [-arb NUM] -pd r1|r2|r3 LOGFILE [MaxRange]
2
```

Or it can take the following instead:

##### Calcdri Alternate Command-line Options

```
1 Calcdri [-arb NUM] -pd r1|r2|r3 LOGFILE [MinRange] [MaxRange]
2
```

The argument **-arb NUM** will specify a numerical precision to run the calculation, it is optional and if omitted it is set to 32. If **-pd** is omitted this script will calculate  $\langle \delta(r_i) \rangle$  using the bash script **CalcExpectation**. In this section the **-pd** option is discussed as it was used throughout this work for all particle densities and is more complex.

The following is a brief guide to how the script operates:

##### Calcdri-Reading the Command-line

```
1 if [[ ${1} == "-arb" ]]
2 then
3     numarb=${2}
4     shift #Shifts the command line arguments along one such that ${2}
           ↪ becomes ${1} and ${3} becomes ${2} and so on.
5     shift
6 fi
7 if [[ ${1} == "-pd" ]] #If omitted calculate  $\langle \delta(r_i) \rangle$  only
8 then
9     pd=1
10    shift
```

### 3. Computer Programming

```
11 fi
12 var=${1}
13 FileName=${2}
14 if [ -n "${4}" ] #if a min and a max range is provided do this
15 then
16     CustomMin=${3}
17     shift
18 fi
19 CustomMax=${3}
20
21 #Checks if the file name is a log or a sv file.
22 if [[ "${FileName}" == */* ]]
23 then
24     rootname="${FileName%*/}/"
25 else
26     rootname=""
27 fi
28 filename="${FileName#*/}"
29 extension="${filename#*.}"
30 filename="${filename%.*}"
31 filename=$(echo "${rootname}${filename}")
32
```

The bash script then prepares the output for appending if the log file was provided. The script will not log to a file if the sv file is given instead of the log file. The sv file was given only during testing whilst the log file was given for final production calculations.

#### Calcdri-Preparing the Files

```
1 savevector=""
2 logfile=""
3 if [ "${extension}" == "log" ] #If it is a log then try to find the sv
  ↪ file
4 then
```

### 3. Computer Programming

```
5  savevector=$(echo "${ffilename}.sv")
6  logfile=$(echo "${FileName}")
7  if [ ! -f "${logfile}" ]
8  then
9      echo "Error log file not found"
10     exit 1
11 fi
12 elif [ ${extension} == "sv" ] #If it is a sv file we will not be
    ↪ logging the data to file this is not recommended
13 then
14     savevector=$(echo "${FileName}")
15     logfile=""
16 else #If it is not a .log or a .sv file then an error is reported
17     echo "The file extention (${extention}) is not .log or .sv"
18     exit 1 #Exit with an error code of 1
19 fi
20
21 if [ ! -f "${svfile}" ]
22 then
23     echo "Error sv file not found"
24     exit 1
25 fi
26 #This block is used to instruct the script only to append the output
    ↪ when the log file is given
27 Append=""
28 if [ -n "${logfile}" ]
29 then
30     Append=$(echo " -a ${logfile}") #Append into the log file
31     echo "" >> "${logfile}"
32 else
33     Append=$(echo "/dev/null") #Append into the null file
34 fi
35
```

### 3. Computer Programming

The next part of the script will prepare the Maple code for calculation. The template Maple script “drmv2.mpl” has several variables unassigned, specifically “ARB”, “SAVEVECTOR” and “DVAR”. These variables are replaced with the desired value using the program `sed` (which is a core utility) and its find and replace feature.

#### Calcdri-Adjusting the Maple Code

```
1 maplefile=$(sed "s|ARB:|${numarb}:|g" "drmv2.mpl")
2 maplefile=$(echo "${maplefile}" | sed
  ↪ "s|SAVEVECTOR|${savevector}|g")
3 maplefile=$(echo "${maplefile}" | sed "s|DVAR|${var}|g")
4 if [ -n "${CustomMin}" ] #Was a different minimum range requested?
5 then
6   maplefile=$(echo "${maplefile}" | sed
  ↪ "s|Min:\=0.0/abs(Z3):|Min:\=${CustomMin}:|g") #Replace the
  ↪ current minimum range with the requested one
7 fi
8 if [ -n "${CustomMax}" ] #Was request a different maximum range
  ↪ requested?
9 then
10   maplefile=$(echo "${maplefile}" | sed
  ↪ "s|Max:\=15/abs(Z3):|Max:\=${CustomMax}:|g") #Replace the
  ↪ current maximum range with the requested one
11 fi
12
```

Finally the prepared Maple code is run using `maple -q` as this enables quiet mode and will not print anything unnecessary. As a final step the pd file that contains the calculated particle densities is loaded and sorted into ascending order of  $r$ . This sorting is necessary because when the Maple code searches for the maximum (see Section 3.3.2) the data is not necessarily calculated in order. Sorting improves readability and some of the plotting tools we use such as `gnuplot` require the data be presorted.

### 3. Computer Programming

#### Calcdri-Calculating with Maple

```
1 maple -q <<< "${maplefile}" | tee ${Append} #Run the Maple code
2 if [ $? -ne 0 ]
3 then
4     exit 1 #If error occurred in maple then quit here
5 fi
6 PDFFILE=$(echo "${ffilename}_dr1.pd")
7 numlines=$(grep -c ^ "${PDFFILE}")
8 numlines=$(echo "${numlines}-2" | bc)
9 SORTED=$(tail -${numlines} "${PDFFILE}" | sort -g) #Load the pd file
    ↪ except the first two lines which are the header lines, and sort
10 HEADER=$(head -2 "${PDFFILE}") #Load the two header lines
11 SORTEDFILE=$(echo -e "${HEADER}\n${SORTED}") #Stick the header and
    ↪ the sorted data back together
12 echo "${SORTEDFILE}" > "${PDFFILE}" #Save the sorted file
13
```

In addition to the command line options `r1`, `r2` and `r3` this script has been extended by another member of the research group to include other more complex densities such as angular densities.<sup>[60]</sup>

#### CalcInnerOuter

The final bash script to be mentioned is `CalcInnerOuter`. All of the expectation values in this script are calculated in Maple. This bash script uses similar techniques to those described in Section 3.4.2 with aspects of the Maple code being changed with `sed` to adjust the parameters prior to calculation. This script was used to calculate all inner and outer expectation values described in Chapter 6 and takes the following arguments.

#### CalcInnerOuter Command-line Options

```
1 CalcInnerOuter r|r^2|r^-1|r1-r2|dr|all LOGFILE
2
```

The command-line arguments `r`, `r2` or `r-1` will calculate the expectations value

### 3. Computer Programming

of  $\langle r_{in}^n \rangle$  and  $\langle r_{out}^n \rangle$  while **dr** will calculate  $\langle \delta(r_{in} - r) \rangle$  and  $\langle \delta(r_{out} - r) \rangle$ . With the argument **dr** extra arguments to control the range are available.

#### CalcInnerOuter Command-line Options For dr

```
1 CalcInnerOuter dr LOGFILE [MaxRange]
2 CalcInnerOuter dr LOGFILE [MinRange] [MaxRange]
3
```

If the argument **all** is given then all the inner outer expectation values expect  $\langle \delta(r_{in} - r) \rangle$  and  $\langle \delta(r_{out} - r) \rangle$  are calculated as these take significantly longer. In this work this script was used to calculate all inner and outer expectation values.

## 3.5. Conclusions

In this chapter the programs written to calculate the data in this thesis are discussed, along with the philosophy and motivation of the programming choices implemented. A combination of programming languages including C++, Maple and bash were used. The C++ programs calculate the eigenvalues, eigenvectors and wavefunction of the various systems and most of the expectation values. The Maple code calculates the recursion relations and the remaining expectation values including the densities and distribution functions. The bash programs are written to automate, simplify and keep the calculations run using C++ and Maple consistent, simple and with clear records. In this chapter we list the various libraries and pieces of code used to form this work and why they were chosen and what they brought to this work. The licenses of these codes are given Appendix F, the most restrictive of which is the numerical recipes license and the MPFR C++ general public license. The C++ code implemented in this work was designed to be flexible, using classes and polymorphism in a modular way. The purpose of which was to reduce the code requirement of future extensions to the program.

# 4. The Non-Relativistic Ground State Energies of Three-body Systems

## 4.1. Introduction

Within Quantum Mechanics the two-body problem is relatively well understood, the problem can be treated exactly in both Schrödinger's<sup>[21]</sup> non-relativistic and Dirac's<sup>[61]</sup> relativistic waveforms. In contrast the three-body problem that introduces an additional particle has no exact solution and is still an active area of research.<sup>[2,8,11,12,28,62]</sup> We use the term three-body (3Body) as a general term to describe systems with three particles interacting via a coulomb potential; these can be any three particle with an arbitrary mass or charge. The third particle can be difficult to deal with; in the case of atoms and atomic ions the third particle is an electron and this introduces complicated electron correlation and exchange. Three-body systems are also under a triangular constraint, this means that the distances between the particles are not independent of each other, although the inter-particle distance is the most convenient way to incorporate electron correlation, this can complicate integration if not treated appropriately. Hylleraas in 1929<sup>[63,64]</sup> provided a method to solve these three-body systems with the explicit inclusion of the inter electron distance, or more generally by explicitly including the interaction of every particle with every other particle. In this method the solution to the Schrödinger equation is solved approximately, however it can still be more accurate than experimental results, as in the case of He which has been solved by Nakashima *et al.*<sup>[65]</sup> to an accuracy of 42 s.f. in Hartrees. This is in excess of experimental values such as the ionisation energy which is known experimentally to be 24.587 41 eV.<sup>[66]</sup>

The variational approach of Hylleraas calls for an approximate wavefunction that



#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

always gives an energy greater than the exact energy of the true wavefunction (details in Section 2.6 on page 11). There have been various different approximations to the wavefunction with different degrees of success. A wavefunction with basis functions that more closely represent the true wavefunction typically have energies that converge faster. Certain terms have been identified as useful for a three-body system in terms of convergence. Bartlett showed that the wavefunction of Hylleraas is not a true solution of the exact wavefunction<sup>[67,68]</sup> and neither are square root terms of the wavefunction and so that these do not improve convergence. Bartlett did however show that natural logarithm terms of the form  $\left(\ln \sqrt{r_1^2 + r_2^2}\right)^p$  are a true solution of the wavefunction and do increase the convergence rate. Indeed it is the logarithmic terms in the works of Schwartz<sup>[62]</sup> and Nakashima *et al.*<sup>[65]</sup> that allow for such highly converged energies. With these wavefunctions however the integrals are harder to calculate and have an increased computational cost. The work of Pekeris<sup>[69–71]</sup> took a different route and by employing an efficient methodology with Laguerre polynomial functions, could achieve high accuracy with more terms rather than fewer better suited terms. This method of Pekeris has been extended to include non-linear variational parameters by various authors.<sup>[72–74]</sup>

As with any method that gives approximate answers, albeit, in this case very accurate and informative approximate answers, the quality of the approximation needs to be explored. The energy of the wavefunction, as with many other mainstream methods is the primary measure by which the quality of the wavefunction is measured. To this purpose, non-linear variational parameters are employed to minimise the energy to the greatest extent possible. This chapter seeks to quantify this process and to ensure that by employing these parameters the ability of the wavefunction to calculate expectation values (See Section 4.4.3) representing physical observables that may be measured, remain just as accurate as the energy of the system under study.

This chapter aims to detail the wavefunction employed in this thesis. Discussing its effectiveness to accurately determine not only energies and properties of known well studied systems but also those that are less studied and more exotic such as muonic atoms and molecules. The mathematical methods used in this thesis to study three-body ground state systems are presented and choices of non-linear variational

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

parameter (NLP) are explained, giving the reasons for including such parameters, when they are best employed, and the effects they have upon a system. The quality of the wavefunction is assessed by calculating a variety of properties in addition to the energy some of which, such as the virial condition, are indicative of wavefunction quality.

## 4.2. Method

This section presents the key theoretical concepts and the methodologies employed. As a general overview, three-body systems were solved with the Schrödinger equation, employing a wavefunction with an exponential term to represent the asymptotic behaviour, and Laguerre polynomials to represent the radial component.

$$\psi(z_1, z_2, z_3) = \sum_{l,m,n=0}^{\infty} C_{l,m,n} e^{-Az_1 - Bz_2 - Cz_3} L_l(Az_1) L_m(Bz_2) L_n(Cz_3) \quad (4.2.1)$$

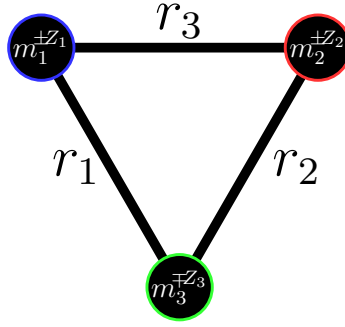
A maximum of 3 non-linear variational parameter denoted in Eq. (4.2.1) as  $A$ ,  $B$  and  $C$  were employed.

### 4.2.1. Perimetric and Triangular Coordinates

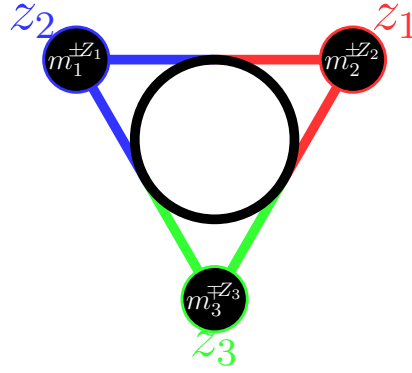
A relatively uncommon set of coordinates, the perimetric coordinates is used (4.2.2). These have several advantages over the more commonly used triangular coordinates that represent the inter-particle distances (Figure 4.2.1). Unlike the triangular coordinates the perimetric coordinates are independent of each other, and range from 0 to infinity. These properties of the perimetric coordinates can greatly reduce the calculation time compared to the inter-particle distance (or triangular) coordinates ( $r_i$ ) and have been used in all calculations. They are however harder to interpret and so in discussions, the triangular coordinates will primarily be referred to. The following conversion can be applied to change between the two.

$$z_i = r_j + r_k - r_i \quad (4.2.2a)$$

$$r_i = \frac{z_j + z_k}{2} \quad (4.2.2b)$$



**Figure 4.2.1.:** Depicts triangular coordinates arranged in this work between three particles of masses  $m_i$  and charges  $Z_i$ .



**Figure 4.2.2.:** Depicts perimetric coordinates,  $z_i$ , the circle is the in-circle of the triangle which partitions the  $r_i$  coordinates to perimetric.

Where  $i, j, k$  are cyclic permutations of 1, 2, 3

#### 4.2.2. Finite Particle Masses

In this work finite masses are used for all particles including the nuclei and other exotic particles. The masses of these particles relative to the electron mass which, is by definition 1, are given in Table 4.2.1.

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

Particle	Mass a.u.
p	1836.152 672
e	1
$\mu$	206.768 284 3
$\tau$	3477.15
d	3670.482 965
t	5496.921 527
He	7294.299 536

**Table 4.2.1.:** These are the finite masses used in this work in atomic units. These values come from the 2010 CODATA of recommend physical constants.<sup>[18]</sup>

### 4.2.3. General Theory

As discussed in Chapter 2 the Schrödinger equation in its time independent form is as follows:

$$\hat{H}\psi = E\psi \quad (4.2.3)$$

$$\hat{H} = \hat{T} + \hat{V} \quad (4.2.4)$$

It is comprised of a kinetic energy part and a potential energy part labelled as  $\hat{T}$  and  $\hat{V}$  respectively. For three-body systems with perimetric coordinates these components take the form in Eqs. (4.2.5) and (4.2.7). Since the s-orbital like states were of primary concern as they represent the ground state, the angular momentum was separated off. The translational motion was separated off to give only the internal energy between the particles. Atomic units were used throughout all calculations unless otherwise stated ( $\hbar = m_e = e = 4\pi\epsilon_0 = 1$ ).

$$\hat{T} = - \sum_{i=1}^3 \left( \nu_{ii} \frac{\partial^2}{\partial z_i^2} + \nu_i \frac{\partial}{\partial z_i} + \sum_{j>i}^3 \nu_{ij} \frac{\partial^2}{\partial z_i \partial z_j} \right) \quad (4.2.5)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

Where

$$\nu_i = 4 \left( -(\mu_i(z_j + z_k))^{-1} - (\mu_j(z_k + z_i))^{-1} - (\mu_k(z_i + z_j))^{-1} \right) \quad (4.2.6a)$$

$$\nu_{ii} = \mu_1^{-1} + \mu_3^{-1} + \mu_3^{-1} - b_{ij}\mu_{ij}^{-1} - b_{ik}\mu_{ik}^{-1} + b_{jk}\mu_{jk}^{-1} \quad (4.2.6b)$$

$$b_{ij} = \frac{2((z_1 + z_2)(z_2 + z_3)(z_3 + z_1) - 2z_i z_j(z_i + z_j))}{(z_1 + z_2)(z_2 + z_3)(z_3 + z_1)} \quad (4.2.6c)$$

$$\nu_{ij} = 2 \left( -\mu_i^{-1} - \mu_j^{-1} - \mu_k^{-1} - b_{ij}\mu_{ij}^{-1} \right) \quad (4.2.6d)$$

$$\mu_i^{-1} = m_i^{-1} + m_3^{-1} \quad i = 1, 2 \quad (4.2.6e)$$

$$\mu_3^{-1} = m_1^{-1} + m_2^{-1} \quad (4.2.6f)$$

$$\mu_{12}^{-1} = m_3^{-1} \quad (4.2.6g)$$

$$\mu_{31}^{-1} = m_1^{-1} \quad (4.2.6h)$$

$$\mu_{23}^{-1} = m_2^{-1} \quad (4.2.6i)$$

$$\hat{V} = \frac{2Z_1 Z_2}{z_1 + z_2} + \frac{2Z_2 Z_3}{z_2 + z_3} + \frac{2Z_3 Z_1}{z_3 + z_1} \quad (4.2.7)$$

When solving the Schrödinger equation variationally the following relationship is observed:

$$\left\langle \psi^{\text{trial}} \mid \hat{H} \mid \psi^{\text{trial}} \right\rangle \geq E^{\text{exact}} \quad (4.2.8)$$

This relationship states that any trial wavefunction will not have an energy lower than the energy that would be obtained with an exact wavefunction. Since an exact wavefunction is not known for three-body coulombic systems, trial wavefunctions are used with the objective of lowering the energy. A prototypical example of solving the Schrödinger equation involves operating the Hamiltonian upon the trial wavefunction, integrating the expression to the form in Eq. (4.2.8), and finding the roots of the equation. Each root of the equation represents a state of the system, where the lowest root is the ground state energy and higher roots represent excited states. This is one of the places where choosing perimetric coordinates is of benefit, since Laguerre polynomials with an exponential term and an integration range of 0 to infinity can be calculated quickly using recursion relations and series solution. When discussing the workings of the calculations in this thesis the discussion will typically be in terms of integrals, however in practise these integrals were instead avoided and

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

evaluated by series solution. This is because series solution is computationally the faster method.

The trial wavefunction used here is made up of a series of basis functions each basis function has the following form:

$$\phi_{lmn}(z_1, z_2, z_3) = e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)} L_l(Az_1) L_m(Bz_2) L_n(Cz_3) \quad (4.2.9)$$

Where the wavefunction is then formed from a sum of these basis functions multiplied by the coefficient,  $C_{lmn}$  and where  $\sum_{l,m,n} C_{lmn}^2 = 1$ .

$$\psi(z_1, z_2, z_3) = \sum_{l,m,n=0}^{\infty} C_{lmn} \phi_{lmn}(z_1, z_2, z_3) \quad (4.2.10)$$

Each Laguerre polynomial function of a particular order has a particular shape such that combining different shapes together with an appropriate coefficient will allow the wavefunction to have whatever shape is necessary to minimise the energy. The appropriate coefficients are calculated by finding the eigenvectors associated with the solutions to the Schrödinger equation. In an ideal world the wavefunction has every possible combination of  $l$ ,  $m$  and  $n$  available from 0 to infinity as this would allow our wavefunction to use any shape from the Laguerre polynomial series. This is not feasible and it is truncated to a finite range from 0 to  $k$ . The more terms that are included the more accurate the energy that can be calculated. However it also takes a significantly longer time.

The  $A$ ,  $B$  and  $C$  in Eq. (4.2.9) are non-linear variational parameters, the wavefunction is in perimetric coordinates and these are harder to interpret than triangular coordinates (See Section 4.2.1). It is often convenient then to discuss the non-linear variational parameters in the triangular coordinates which are simply the distances between particles. We denote  $\alpha$ ,  $\beta$  and  $\gamma$  to represent the non-linear variational parameters when using triangular coordinates such that the exponential transforms as:

$$e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)} = e^{-(\alpha r_1+\beta r_2+\gamma r_3)} \quad (4.2.11)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

This is achieved using the following conversions:

$$\alpha = \frac{B + C - A}{2} \quad (4.2.12)$$

$$\beta = \frac{C + A - B}{2} \quad (4.2.13)$$

$$\gamma = \frac{A + B - C}{2} \quad (4.2.14)$$

To convert in the reverse direction the following transformation is applied:

$$A = \beta + \gamma \quad (4.2.15)$$

$$B = \gamma + \alpha \quad (4.2.16)$$

$$C = \alpha + \beta \quad (4.2.17)$$

#### 4.2.4. Laguerre Polynomials

The Laguerre polynomials ( $L$ ) are a series of polynomials that are solutions of the following expression:

$$xy'' + (1 - x)y' + my = 0 \quad (4.2.18)$$

Eq. (4.2.18) is satisfied when:  $y = L_m(x)$ . The first four Laguerre polynomials have the following form:

$$L_0(x) = 1 \quad (4.2.19)$$

$$L_1(x) = 1 - x \quad (4.2.20)$$

$$L_2(x) = 1 - 2x + \frac{1}{2}x^2 \quad (4.2.21)$$

$$L_3(x) = 1 - 3x + \frac{3}{2}x^2 - \frac{1}{6}x^3 \quad (4.2.22)$$

Each individual Laguerre polynomial can be generated using either of these following forms for  $L_m(x)$  (where  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ ). This is not an exhaustive list of generating

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

functions for Laguerre polynomials:<sup>[75–78]</sup>

$$L_m(x) = \sum_{i=0}^m (-1)^i \binom{m}{m-i} \frac{x^i}{i!} \quad (4.2.23)$$

$$L_m(x) = \frac{e^x}{m!} \frac{\partial^m}{\partial x^m} (e^{-x} x^m) = \frac{1}{m!} \left( \frac{\partial}{\partial x} - 1 \right)^m x^m \quad (4.2.24)$$

The Laguerre polynomials were chosen as the radial component of the wavefunction since they are an orthogonal set and range from 0 to infinity as do the coordinates of the chosen coordinate system. The Laguerre polynomials orthogonality relationship is given in Eq. (4.2.25) and requires that the Laguerre polynomial be multiplied by an exponential term. This is convenient since exponential terms were used to represent the asymptotic part of the wavefunction as they “kill” the wavefunction appropriately at long range i.e. when the inter-particle distances become large.

$$\int_0^\infty e^{-x} L_m(x) L_n(x) dx = \delta_{m,n} = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{if } m \neq n \end{cases} \quad (4.2.25)$$

To solve the Schrödinger equation the wavefunction was operated upon and integrated, (in practise this was converted to a series solution for rapid evaluation). The above orthogonality relationship was used to integrate these expressions. This makes the integration simple, as the expression evaluated to either 1 or 0. Although when operating  $\hat{H}$  on the wavefunction used here, terms such as  $e^{-x} L'_m(x) L_n(x)$  are generated that do not fit this relation and without further work would be difficult to integrate. In Section 4.2.5 a method to resolve these forms of expressions such that they fit the orthogonality relationship is presented (recurrence relations). Therefore we can use this to speed up the calculation for the Hamiltonian and many other operators. This relationship is beneficial in terms of simplicity and computational time therefore the wavefunction is adjusted only in ways that did not lose this relationship, leading to a consequence that will be discussed later in Section 4.2.9.



#### 4.2.5. Recurrence Relations & Series Solution

The Laguerre polynomials have a set of recurrence relations that can simplify the calculations, these are:<sup>[75-77]</sup>

$$xL_n(x) = -(n+1)L_{n+1}(x) + (2n+1)L_n(x) - nL_{n-1}(x) \quad (4.2.26a)$$

$$xL'_n(x) = nL_n(x) - nL_{n-1}(x) \quad (4.2.26b)$$

$$xL''_n(x) = (x-1)L'_n(x) - nL_n(x) \quad (4.2.26c)$$

From Eqs. (4.2.5) and (4.2.7), operating the Hamiltonian upon the wavefunction ( $\hat{H}\psi$ ), results in an expression containing both first and second derivative Laguerre polynomials. This  $\hat{H}\psi$  causes Laguerre polynomial terms to appear that do not conform to the orthogonality relationship in Eq. (4.2.25) (e.g.  $e^{-x}L'_m(x)L_n(x)$ ). By applying the recurrence relations given in Eqs. (4.2.26a) to (4.2.26c) to  $\hat{H}\psi$  the second and first derivatives etc. can be converted to a form that conforms with the orthogonality relationship and be easily integrated to either 1 or 0. This greatly simplified the formation of the matrix that represents the problem. To improve the calculation of this matrix further we can combine the recurrence relations to form a new recurrence relation that represents the Schrödinger equation, by picking the appropriate expression based upon the  $l$ ,  $m$  and  $n$  indices. This process of picking the expression is the series solution method and is an efficient method of the matrix formation without performing integration explicitly.

Take for example the integral  $\langle r_3 \rangle$  this can be solved using series solution in the following way. For simplicity the non-linear variational parameters have been set to 1 and we ignore the Jacobian of transformations in this discussion.

$$\int_0^\infty \int_0^\infty \int_0^\infty \psi^* r_3 \psi dz_1 dz_2 dz_3 \quad (4.2.27)$$

Applying the conversion in Eq. (4.2.2b) to Eq. (4.2.27)

$$\int_0^\infty \int_0^\infty \int_0^\infty \psi^* \frac{z_1 + z_2}{2} \psi dz_1 dz_2 dz_3 \quad (4.2.28)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

Substituting in Eq. (4.2.10)

$$\begin{aligned} & \int_0^\infty \int_0^\infty \int_0^\infty \left( \sum_{lmn=0}^\infty C_{lmn} \phi_{lmn}(z_1, z_2, z_3) \right) \frac{z_1 + z_2}{2} \\ & \times \left( \sum_{l'm'n'=0}^\infty C_{l'm'n'} \phi_{l'm'n'}(z_1, z_2, z_3) \right) dz_1 dz_2 dz_3 \end{aligned} \quad (4.2.29)$$

These equations are processed in matrix form where each individual matrix element  $(M_{i,j})$  has the following form, where  $i$  and  $j$  are indices

$$M_{i,j} = \int_0^\infty \int_0^\infty \int_0^\infty \phi_{lmn}(z_1, z_2, z_3) \frac{z_1 + z_2}{2} \phi_{l'm'n'}(z_1, z_2, z_3) dz_1 dz_2 dz_3 \quad (4.2.30)$$

The  $\phi_{lmn}$  expands into the following using Eq. (4.2.9) and some minor simplifications

$$\begin{aligned} M_{i,j} = & \frac{1}{2} \int_0^\infty \int_0^\infty \int_0^\infty e^{-(z_1+z_2+z_3)} L_l(z_1) L_m(z_2) L_n(z_3) (z_1 + z_2) \\ & \times L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) dz_1 dz_2 dz_3 \end{aligned} \quad (4.2.31)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

This is then expanded before applying the recursion relations of Eq. (4.2.26)

$$M_{i,j} = \frac{1}{2} \int_0^\infty \int_0^\infty \int_0^\infty e^{-(z_1+z_2+z_3)} (L_l(z_1) L_m(z_2) L_n(z_3) z_1 L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) \\ + L_l(z_1) L_m(z_2) L_n(z_3) z_2 L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3)) \\ dz_1 dz_2 dz_3 \quad (4.2.32)$$

$$M_{i,j} = \frac{1}{2} \int_0^\infty \int_0^\infty \int_0^\infty e^{-(z_1+z_2+z_3)} ((- (l+1) L_{l+1}(z_1) + (2l+1) L_l(z_1) \\ - l L_{l-1}(z_1)) L_m(z_2) L_n(z_3) L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) \\ + L_l(z_1) (- (m+1) L_{m+1}(z_2) + (2m+1) L_m(z_2) \\ - m L_{m-1}(z_2)) L_n(z_3) L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3)) \\ dz_1 dz_2 dz_3 \quad (4.2.33)$$

$$M_{i,j} = \frac{1}{2} \int_0^\infty \int_0^\infty \int_0^\infty e^{-(z_1+z_2+z_3)} (- (l+1) L_{l+1}(z_1) L_m(z_2) L_n(z_3) L_{l'}(z_1) \\ \times L_{m'}(z_2) L_{n'}(z_3) + (2l+1) L_l(z_1) L_m(z_2) L_n(z_3) \\ \times L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) - l L_{l-1}(z_1) L_m(z_2) L_n(z_3) \\ \times L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) - L_l(z_1) (m+1) L_{m+1}(z_2) \\ \times L_n(z_3) L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) + L_l(z_1) (2m+1) \\ \times L_m(z_2) L_n(z_3) L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3) - L_l(z_1) m \\ \times L_{m-1}(z_2) L_n(z_3) L_{l'}(z_1) L_{m'}(z_2) L_{n'}(z_3)) dz_1 dz_2 dz_3 \quad (4.2.34)$$

By then applying the orthogonality relation of Eq. (4.2.25) we solve the series

$$M_{i,j} = \frac{1}{2} (- (l+1) \delta_{l+1,l'} \delta_{m,m'} \delta_{n,n'} + (2l+1) \delta_{l,l'} \delta_{m,m'} \delta_{n,n'} - l \delta_{l-1,l'} \delta_{m,m'} \delta_{n,n'} \\ - (m+1) \delta_{l,l'} \delta_{m+1,m'} \delta_{n,n'} + (2m+1) \delta_{l,l'} \delta_{m,m'} \delta_{n,n'} - m \delta_{l,l'} \delta_{m-1,m'} \delta_{n,n'}) \quad (4.2.35)$$

Equation (4.2.35) is the generic form for the entire series for our wavefunction. In this case it has 6 terms but different operators have a different number of terms.

The matrix is formed with certain variables such as  $m_i$  and  $Z_i$  unassigned to create a generic reusable solution. The indices of the matrix  $i$  and  $j$  are converted into  $l, m, n$  and  $l', m', n'$  using the collapsed index scheme in Appendix A.

##### 4.2.6. Machine Precision and Numerical Stability

Using Laguerre polynomials to form the wavefunction allowed for many simplifications of the calculation whilst retaining high accuracy. Additionally since the integration is either one or zero, there are lots of terms in the matrix that become zero. When a matrix has mainly zero elements it is known as a sparse matrix and these can be numerically stable, since operating on zero elements does not introduce rounding error. A numerically stable matrix is useful as it allows for many more terms in the wavefunction without introducing machine error. Within most modern computers IEEE 754 (Institute of Electrical and Electronics Engineer)<sup>[79]</sup> is the standard for computer data types. On standard compilers the most accurate built-in data type is typically the double precision data type that has a maximum precision of 16 s.f.. If an ill conditioned matrix is calculated i.e. where its condition number (the logarithm of ratio between the largest to smallest eigenvalue) is larger than the precision of the elements then results are unreliable.<sup>[80]</sup> The precision error is usually resolved by using specially designed data types that have more significant figures. Since the matrix used here is sparse it is numerically stable and is less likely to have an error introduced by the machine precision even with large wavefunctions. The process used to generate our wavefunctions through the Schrödinger equation was numerically stable and the use of arbitrary precision, which can be very slow, was not necessary. Instead the fast built-in types were used (in a later chapter it will be shown that this is not sufficient for all operators). To take advantage of these benefits of the Laguerre polynomials it was ensured at all times that the wavefunction basis remains orthogonal so that Eq. (4.2.25) holds true.

With this method it is ensured that the process of solving the eigenproblem and calculating the wavefunction is mathematically stable. Certain mathematical operators however can be numerically unstable<sup>[81]</sup> requiring greater precision to accurately calculate after the stable solution of the wavefunction. The Dirac deltas discussed in Section 6.2.2 on page 173 are an example of this.

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

As a result of this some features known to benefit three-body systems were not implemented such as the logarithmic terms that have been proven to be useful<sup>[62,68,82,83]</sup> for three-body systems. Later it will also be shown that it also limits our choice of non-linear variational parameters when considering fermionic exchange (Section 4.2.9).

##### 4.2.7. Asymptotic Behaviour

**The Two-body Asymptotic Behaviour** During this chapter the asymptotic nature of the wavefunction will be referred back to at several points, and for this reason it is useful to define the two particle asymptotic nature and how this can be extended with some limitations to three-body systems. When this two particle asymptotic nature is extended to the three particle system it is known as the formal solution<sup>[15–17]</sup> and was used by Hylleraas<sup>[63,64]</sup> and other authors such as Pekeris<sup>[69]</sup> to form their exponential term. A discussion of this solution can be found in Bransford Joachain<sup>[84]</sup> and Bethe and Salpeter.<sup>[17]</sup>

The following discussion explains briefly how the two particle asymptotic condition can be derived. When considering the ground state of a two-body time independent system, the wavefunction can be fully represented by the inter-particle distance ( $r$ ). In this case the Schrödinger equation has the following form (where  $E_{2b}$  is the two-body energy):

$$\hat{H}(r)\psi = E_{2b}\psi(r) \quad (4.2.36)$$

This is used to derive the asymptotic condition i.e. where  $r$  tends towards infinity  $\psi(r)$  becomes the asymptotic solution. In this discussion the hydrogenic system with atomic units and infinite nuclear mass is used.

$$(\hat{H} - E_{2b})\psi(r) = 0 \quad (4.2.37)$$

$$\hat{H} = \hat{T} + \hat{V} \quad (4.2.38)$$

$$\hat{V} = -\frac{Z}{r}, \quad (4.2.39)$$

$$\hat{T} = -\frac{1}{2} \frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} \quad (4.2.40)$$

In the asymptotic region  $r$  tends towards infinity and when this occurs  $\hat{T}$  and  $\hat{V}$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

change in the following way:

$$\text{as } r \rightarrow \infty, \hat{V} \rightarrow 0, \hat{T} = -\frac{1}{2} \frac{\delta^2}{\delta r^2} \quad (4.2.41)$$

As a result the Schrödinger equation simplifies to:

$$-\frac{1}{2} \frac{\delta^2 \psi(r)}{\delta r^2} - E_{2b} \psi(r) = 0 \quad (4.2.42)$$

Solving this gives the asymptotic form. Generally the equation to be solved has the form:

$$ay''(x) + by'(x) + cy(x) = 0 \quad (4.2.43)$$

Where  $a$ ,  $b$ , and  $c$  are constants and the equation has distinct roots.<sup>[76,85]</sup> In this case we have the following situation:

$$ay''(r) + cy(r) = 0 \quad (4.2.44)$$

$$a = -\frac{1}{2} \quad (4.2.45)$$

$$c = -E_{2b} \quad (4.2.46)$$

Let  $D$  denote the derivative and  $D^2$  denote the second derivative and then solve as a characteristic equation<sup>[76,85]</sup>

$$D = \frac{\delta}{\delta r} \quad (4.2.47)$$

$$-\frac{1}{2} D^2 y(r) - E_{2b} y(r) = 0 \quad (4.2.48)$$

$$\left(-\frac{1}{2} D^2 - E_{2b}\right) y(r) = 0 \quad (4.2.49)$$

$$(-D^2 - 2E_{2b}) y(r) = 0 \quad (4.2.50)$$

$D$  is therefore:

$$-D^2 = 2E_{2b} \quad (4.2.51)$$

$$D^2 = -2E_{2b} \quad (4.2.52)$$

$$D = \pm \sqrt{-2E_{2b}} \quad (4.2.53)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

Therefore  $Dy(r)$  is:

$$Dy(r) = \pm \sqrt{-2E_{2b}}y(r) \quad (4.2.54)$$

Since  $Dy(r)$  is the differential of  $y(r)$  then it follows that the integral of  $Dy(r)$  is  $y(r)$

$$\frac{\delta y(r)}{\delta r} = \pm \sqrt{-2E_{2b}}y(r) \quad (4.2.55)$$

$$\frac{1}{y(r)} \frac{\delta y(r)}{\delta r} = \pm \sqrt{-2E_{2b}} \quad (4.2.56)$$

$$\int \frac{1}{y(r)} \delta y(r) = \int \pm \sqrt{-2E_{2b}} \delta r \quad (4.2.57)$$

$$\ln y(r) = \pm \sqrt{-2E_{2b}}r + c \quad (4.2.58)$$

At  $r = \infty$ , then  $y(r) = 0$  and therefore  $c$  must be 0

$$y(r) = e^{\pm \sqrt{-2E_{2b}}r} \quad (4.2.59)$$

$$y(r) = \psi(r) \quad (4.2.60)$$

$$\psi(r) = e^{\pm \sqrt{-2E_{2b}}r} \quad (4.2.61)$$

As the positive solution is physically inaccurate for bound coulombic systems and energy is negative, this gives us the exact asymptotic nature of the two-body system to be:

$$\psi(r) = e^{-\sqrt{-2E_{2b}}r} \quad (4.2.62)$$

In the case of the two-body system the energy  $E_{2b}$  for atomic systems is known exactly as:

$$E_{2b} = E_{ij} = -\frac{Z_i^2 Z_j^2}{2n^2} \frac{m_i m_j}{m_i + m_j} \quad (4.2.63)$$

For an atomic two-body system with infinite nuclear mass and a single electron this becomes:

$$E_{2b} = -\frac{Z^2}{2n^2} \quad (4.2.64)$$

$Z$  is the nuclear charge and  $n$  is the principle quantum number. In the ground state where  $n = 1$  this asymptote becomes:

$$\psi(r) = e^{-Zr} \quad (4.2.65)$$

It is worth noting that the exponent is only  $-Zr$  within the infinite nuclear mass approximation, for other masses where  $m_1, m_3$  are the masses of oppositely charged

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

particles with charges  $Z_1$  and  $Z_3$  it is:

$$\psi(r) = e^{-\sqrt{\frac{m_1 m_3}{m_1 + m_3}} |Z_1 Z_3| r} \quad (4.2.66)$$

**The Three-body Asymptotic Behaviour** Extending this to 3 body systems is more difficult as the repulsive interaction between the two like-charged particles is difficult to deal with. It is possible to extend the two-body solution only when ignoring this repulsive interaction, and assuming the non-interacting particle model between the like-charged particles. There are 4 asymptotic regions for three-body systems,<sup>[8]</sup> and here the same notation used in other papers will be used.<sup>[15,16]</sup>  $\Omega_0$  denotes the first asymptotic region where all inter-particle distances tend to infinity. In this region the independent particle approximation would be valid since any particles would be too far apart for interaction;  $\Omega_1$ ,  $\Omega_2$ ,  $\Omega_3$  denote the other 3 regions and they represent the systems where only one particle tends towards infinity, with  $\Omega_1$  for particle 1 tending towards infinity  $\Omega_2$  for particle 2 and  $\Omega_3$  for particle 3.

**The  $\Omega_0$  Region** The region best described is that of  $\Omega_0$  since the particles are not interacting in this region. This allows for some useful simplifications and approximations; (i), the interactions between the like-charge particles is negligible and the inter-particle coordinate between the 2 like-charge particles can be ignored. For this discussion let this coordinate be  $r_3$ ; (ii), the Hamiltonian becomes the sum of two, two-body Hamiltonians one for the  $r_1$  coordinate and one for the  $r_2$  coordinate;<sup>[84]</sup> (iii), the energy becomes the sum of two, two-body energies, one for  $r_1$  and one for  $r_2$ .<sup>[84]</sup> With these approximations we can calculate the asymptotic nature for this region, in exactly the same way as the two-body system.

$$\left( \hat{H} - E_{3b} \right) \psi(r_1, r_2) = 0 \quad (4.2.67)$$

When assuming no interaction along the  $r_3$  coordinate, the energy becomes the sum of two, two-body energies

$$E_{3b} = E_{13} + E_{23} \quad (4.2.68)$$

where  $E_{ij}$  is the two-body  $E_{2b}$  energy (See Eq. (4.2.63)) corresponding to the interaction between particles connected by that coordinate i.e. for the oppositely



#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

charged particles connected by  $r_1$  the energy is  $E_{13}$  and for those connected by  $r_2$  it is  $E_{23}$ . The Hamiltonian is also the sum of two, two-body Hamiltonians. When all inter-particle distances tend towards infinity this Hamiltonian reduces to the kinetic energy, as in the two-particle case.

$$\text{as } r_1 \rightarrow \infty \text{ and } r_2 \rightarrow \infty \quad (4.2.69)$$

$$\hat{V} \rightarrow 0 \text{ and } \hat{T} = -\frac{1}{2} \frac{\delta^2}{\delta r_1^2} - \frac{1}{2} \frac{\delta^2}{\delta r_2^2} \quad (4.2.70)$$

$$\left( -\frac{1}{2} \frac{\delta^2}{\delta r_1^2} - \frac{1}{2} \frac{\delta^2}{\delta r_2^2} - (E_{13} + E_{23}) \right) \psi(r_1, r_2) = 0 \quad (4.2.71)$$

Applying the same characteristic equations as before yields:

$$\psi(r_1, r_2) = e^{-\sqrt{-2E_{13}}r_1 - \sqrt{-2E_{23}}r_2} \quad (4.2.72)$$

This can be further simplified by assuming that the interaction of the two pairs of attracting particles is the same, i.e.  $E_{13} = E_{23}$  this is valid for the two electron case in the ground state when assuming no repulsive interaction. The expression then becomes:

$$\psi(r_1, r_2) = e^{-\sqrt{-2E_{13}}(r_1+r_2)} \quad (4.2.73)$$

This is in principle an exact form of the three-body asymptotic behaviour, for the region  $\Omega_0$ .

The two-body energy ( $E_{13}$ ) can be expanded using Eq. (4.2.63) to give the full particles in motion solution:

$$E_{13} = -\frac{Z_1^2 Z_3^2}{2n^2} \frac{m_1 m_3}{m_1 + m_3} \quad (4.2.74)$$

$$e^{-\sqrt{2E_{13}}(r_1+r_2)} = e^{-\sqrt{\frac{Z_1^2 Z_3^2}{n^2} \frac{m_1 m_3}{m_1 + m_3}}(r_1+r_2)} \quad (4.2.75)$$

By applying the conditions for an infinite nuclear mass atom i.e. where  $n = 1$ ,  $Z_1 = -1$ ,  $Z_3 = Z$ ,  $m_1 = 1$ ,  $m_3 = \infty$ . Then the fixed particle hydrogenic case is acquired as:

$$\psi(r_1, r_2) = e^{-Z(r_1+r_2)} \quad (4.2.76)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

Some authors have found that when adding a NLP into the exponential term representing the asymptote, such a parameter is optimal when it approximately equals  $Z$ .<sup>[84]</sup> This asymptotic form used by Pekeris has several limitations, firstly the asymptotic form of the wavefunction is only valid in the  $\Omega_0$  region where all the particles are infinitely far apart. This means that it is not valid for the other regions, where only one particle is infinitely far apart from the other two (See work by Kim *et al.*<sup>[15]</sup> and Alt *et al.*<sup>[16]</sup> for attempts to extend to all regions). When the majority of the mass is on a single particle and the other two particles are the same, such as in He, the system behaves much like the two-body asymptote upon which this asymptotic form is based, i.e. when 2 particles can freely escape to infinity, in these cases the formal solution should be a good approximation. In a case where two particles are much heavier and one much lighter, such as in  $\text{H}_2^+$ , the asymptotic form here would not be expected to perform well (this is shown in Section 4.4.1) since the two protons would not be expected to move much relative to the electron. As a result it is much harder to separate all particles uniformly to infinity. Additionally the  $r_3$  (this is the proton–proton distance) interaction would be expected to be more significant.

The last limitation of our asymptotic form is that we assumed that  $E_{13} = E_{23}$ . This however is not always the case, particularly if, for example, one electron is in a higher excited state. To demonstrate this, consider the full two-body (2Body) energy for an atom with infinite nuclear mass.

$$E_{2b} = -\frac{Z^2}{2n^2} \quad (4.2.77)$$

If the first electron is in the  $n = 1$  ground state and the second in the  $n = 2$  excited state then  $E_{13} \neq E_{23}$ . In this case the asymptotic condition would instead have the full form of:

$$\psi(r_1, r_2) = e^{-\sqrt{\frac{Z^2}{2 \times 1^2}} r_1 - \sqrt{\frac{Z^2}{2 \times 2^2}} r_2} \quad (4.2.78)$$

#### 4.2.8. Non-Linear Variational Parameters: The Wavefunction Forms

The non-linear variational parameters in the wavefunction were chosen specifically to alter the exponential of the wavefunction to increase the rate of convergence,

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

whilst keeping the possibility of the system to choose the formal solution. Therefore in this report the parameters will be referred to as being part of the asymptote. However, as mentioned earlier, they appear in the definition of the coordinate system. This exponential must always have a negative exponent to satisfy the orthogonality relationship of Eq. (4.2.25), but also to ensure that the wavefunction does “die off” at long range. When a particle has sufficient distance from the other two particles it becomes unbound. This means that the exponent affects the asymptotic nature of the wavefunction. Therefore appropriate non-linear variational parameters in the exponent alter the point at which the particle becomes unbound. The original wavefunction by Pekeris<sup>[71]</sup> had an exponent of the form:

$$e^{-\sqrt{E}(r_1+r_2)} \quad (4.2.79)$$

As discussed this is the formal solution for the  $\Omega_0$  region with symmetric systems when assuming no interaction on the  $r_3$  coordinate. There are no non-linear variational parameters, and the asymptote has no dependence on the  $r_3$  coordinate. The wavefunction used by Cox *et al.* has an exponential of the form:

$$e^{-\sqrt{\frac{E}{K}}(r_1+r_2)} \quad (4.2.80)$$

Both of these forms of the wavefunction were shown to give good energy convergence for atom-like systems.<sup>[69–71,73,86,87]</sup> However for molecule-like systems the convergence of the energy was not as pronounced.<sup>[88]</sup> Both of these wavefunction have been implemented within this project and their performance is discussed later. In this work the following three exponentials forms are implemented:

$$e^{-\alpha(r_1+r_2)} \quad (4.2.81)$$

$$e^{-\alpha(r_1+r_2)-\gamma r_3} \quad (4.2.82)$$

$$e^{-\alpha r_1-\beta r_2-\gamma r_3} \quad (4.2.83)$$

In Eq. (4.2.76) it can be seen that the asymptote for the formal solution is directly related to the charge of nucleus. The exponent in Eq. (4.2.81) was chosen to investigate this property. The wavefunction of Drake *et al.*'s<sup>[89]</sup> had an exponent with a

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

form similar to this as  $e^{-\alpha r_1 - \beta r_2}$ , where they showed that in the Helium ground state  $\alpha$  has an optimum value of the charge ( $Z$ ) and  $\beta$  an optimum value of charge minus one (effective charge), which is the screened charge. This form was not possible for symmetric systems in this methodology as will be discussed later, however it does show for the atomic systems Drake investigated, that the exponential form has a dependence on the nuclear charge as suggested by the formal solution. The exponent in Eq. (4.2.81) still assumes that the interaction between the like-charged particles is negligible. The exponent in Eq. (4.2.82) extends this by introducing this extra  $r_3$  coordinate into the asymptote; this was chosen to investigate  $r_3$  dependence on the asymptote, particularly for systems where this coordinate is expected to be more important such as in molecular systems. The exponent in Eq. (4.2.83) is suitable for systems without fermionic exchange for reason explained in Section 4.2.9. This form separates the asymptote of the like-charged particles.

#### 4.2.9. The Effects of Fermionic Symmetry

When two particles are identical and have half integer spin the symmetry of the system must be taken into account. Since two identical particles with half integer spin undergo a process known as fermionic exchange<sup>[17,84]</sup> where fermionic particles are antisymmetric with respects to exchange. To accommodate this the trial wavefunction takes on the following form where the  $z_1$  and  $z_2$  coordinates have been interchanged:

$$\psi(z_1, z_2, z_3) = \psi(z_1, z_2, z_3) \pm \psi(z_2, z_1, z_3) \quad (4.2.84)$$

This expands to the following:

$$\begin{aligned} \psi(z_1, z_2, z_3) \pm \psi(z_2, z_1, z_3) = & \sum_{lmn=0}^{\infty} C_{lmn} \left( e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)} L_l(Az_1) L_m(Bz_2) \right. \\ & \times L_n(Cz_3) \pm (1 - \delta_{lm}) e^{-\frac{1}{2}(Az_2+Bz_1+Cz_3)} L_l(Az_2) \\ & \left. L_m(Bz_1) L_n(Cz_3) \right) \end{aligned} \quad (4.2.85)$$

When the sign is positive it is the symmetric case and when it is negative it is the anti-symmetric case. States such as the  $^1S$  state are symmetric. By exploiting the symmetry the number of terms in the wavefunction can be reduced. The symmetry

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

rules lead to some of the eigenvalues being either degenerate, or zero. This has been taken into account and as a result the symmetric and antisymmetric wavefunctions have fewer terms in them than the asymmetric wavefunctions for the same Pekeris shell (See Section 4.2.10).

**Why the  $\beta$  parameter and fermionic exchange causes the loss of orthogonality** When incorporating the  $\beta$  parameter into the symmetric wavefunction, the orthogonality is lost. This can be seen when taking the symmetric basis function:

$$\begin{aligned} & e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)} L_l(Az_1) L_m(Bz_2) L_n(Cz_3) \\ & + e^{-\frac{1}{2}(Az_2+Bz_1+Cz_3)} L_l(Az_2) L_m(Bz_1) L_n(Cz_3) \end{aligned} \quad (4.2.86)$$

and multiplying by itself to simulate  $\psi^*\psi$ :

$$\begin{aligned} & \left( e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)} L_l(Az_1) L_m(Bz_2) L_n(Cz_3) \right. \\ & \left. + e^{-\frac{1}{2}(Az_2+Bz_1+Cz_3)} L_l(Az_2) L_m(Bz_1) L_n(Cz_3) \right)^2 \\ & = e^{-(Az_1+Bz_2+Cz_3)} L_l(Az_1)^2 L_m(Bz_2)^2 L_n(Cz_3)^2 \\ & \quad + e^{-(Bz_1+Az_2+Cz_3)} L_l(Az_2)^2 L_m(Bz_1)^2 L_n(Cz_3)^2 \\ & \quad + 2e^{-\frac{1}{2}((A+B)z_1+(B+A)z_2)-Cz_3} L_l(Az_1) L_l(Az_2) L_m(Bz_2) L_m(Bz_1) L_n(Cz_3)^2 \end{aligned} \quad (4.2.87)$$

The last term in Eq. (4.2.87) no longer satisfies the Laguerre polynomial orthogonality relationship (Equation (4.2.25)) since the expression inside the Laguerre polynomial terms for  $z_1$  and  $z_2$  ( $L_l(Az_1) L_m(Bz_1)$  and  $L_l(Az_2) L_m(Bz_2)$ ) differ from each other due to different non-linear variational parameters and those in the exponent  $e^{-\frac{1}{2}(A+B)z_1}$ , resulting in an expression that does not resemble the orthogonality relationship. However when  $A = B$ , orthogonality is maintained. As a result of this it is not possible to include an exponent with a  $B$  parameter in systems with fermionic exchange whilst retaining the orthogonality relationship and series solution. The non-linear variational parameter  $B$  can be used however in systems without this exchange, such as  $\text{HD}^+$ . when all particles differ from each other. In these asymmetric cases all 3 non-linear variational parameters can be used with the wavefunction shown in Eq. (4.2.83).

$l$	$m$	$n$	$\omega$
0	0	0	0
0	0	1	1
0	1	0	1
0	0	2	2
0	1	1	2
0	2	0	2
1	1	0	2

**Table 4.2.2.:** The first few Pekeris shells for symmetric systems

#### 4.2.10. Pekeris Shell

Within this thesis the term Pekeris shell is on occasion used to define the number of terms in the trial wavefunction. When the number of terms in the trial wavefunction is increased it is done in whole Pekeris shell number. Each Pekeris shell ( $\omega$ ) defines a set of  $l$ ,  $m$  and  $n$  for a wavefunction where only  $l + m + n \leq \omega$  are included in the trial wavefunction. For further detail refer to the original paper by Pekeris *et al.*<sup>[71]</sup> As an example Table 4.2.2 lists the first few Pekeris shells and their associated  $l$ ,  $m$  and  $n$  values for the symmetric case.

The number of terms in a complete Pekeris shell may be generated with the following equations:

Number of terms asymmetric wavefunction

$$k_{asym} = 1 + \frac{11}{6}\omega + \omega^2 + \frac{1}{6}\omega^3 \quad (4.2.88)$$

Number of terms symmetric wavefunction

$$k_{sym} = \frac{15}{16} + \frac{17}{12}\omega + \frac{5}{8}\omega^2 + \frac{1}{12}\omega^3 + \frac{1}{16}(-1)^\omega \quad (4.2.89)$$

Number of terms antisymmetric wavefunction

$$k_{antisym} = \frac{15}{16} + \frac{17}{12}(\omega - 1) + \frac{5}{8}(\omega - 1)^2 + \frac{1}{12}(\omega - 1)^3 + \frac{1}{16}(-1)^{(\omega-1)} \quad (4.2.90)$$

### 4.3. Implementation

In this Chapter ground state variationally time independent energies were calculated using the series solution method as described in detail in Section 4.2.5. The wavefunction was formed from a series of Laguerre polynomials in perimetric coordinates

(Equation (4.2.1)). The coefficients of the wavefunction were determined by solving the generalised eigenvalue equation. This was done by first forming a general matrix using series solution that represents the integral  $\int_0^\infty \int_0^\infty \int_0^\infty \psi^* \hat{H} \psi \, dz_1 \, dz_2 \, dz_3$ . By general matrix it is meant that  $m_i$ 's,  $Z_i$ 's are left undetermined at said stage. Then by using Cholesky Decomposition, followed by Householder reduction<sup>[29,30]</sup> and then solving the eigenvalue problem, the eigenvalues and eigenvectors were acquired using multiple relatively robust representations (MR3).<sup>[90,91]</sup> The non-linear variational parameters were optimised to a minimum using the methods Numerical Recipes optimisation (NROpt) and bound optimisation by quadratic approximation (BOBYQA) as described in Section 3.2.1 on page 35.

## 4.4. Results and Discussion

### 4.4.1. The Effect of the Non-Linear Variational Parameters

When the number of basis functions in a trial wavefunction is increased the energy decreases. The energy of a variational calculation, as employed here, can never go below the true energy that would be acquired with an exact wavefunction.<sup>[17,84,92]</sup> Each additional basis function can be thought of as a corrective term that allows the wavefunction to achieve a “shape” that better describes the system. The better a wavefunction form describes the system, the fewer basis functions are required to achieve a given energy. For this reason using non-linear variational parameters can, with an appropriate value, allow the calculation of better energies for the system with fewer basis functions and therefore in less time. With any trial wavefunction a point will be reached whereby increasing the number of basis functions does not decrease the energy calculated significantly. The definition of “significantly” depends on the desired accuracy. At this point it is possible to say that the energy is converged. This is one of the measures by which we compare our wavefunctions; a better wavefunction will reach this convergence point with fewer basis functions.

In this section energies are presented that were calculated using each of the wavefunctions described above for various three-body systems as a function of the Pekeris shell. The rate of convergence is studied for each of the wavefunctions. Different systems such as atomic, molecular and exotic are expected to converge to different

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

degrees for different forms of the wavefunction. An atomic system is defined as one where the uniquely charged particle is much heavier than the other two. A molecular system is defined as one with the two identically charged particles much heavier than the uniquely charged particle; and the term “exotic” is used for the other systems in between these extremes. They are named “exotic” as these systems usually involve exotic particles such as muons ( $\mu$ ). In each of the following tables the digits that are in bold are considered converged. Converged is defined as either matching to the best literature data available or matching to the best higher order calculation when literature data are not available. The benefits of including the non-linear variational parameters on the convergence will be studied.

The following notations are used to describe each wavefunction and all of these have been implemented in code in this work.

$\psi_{\text{pekeris}}$  refers to the original Pekeris wavefunction with the exponential form in Eq. (4.4.1).<sup>[69–71]</sup> The asymptote includes the  $\sqrt{E}$ . Unlike in the other wavefunction we will discuss this wavefunction has no non-linear variational parameters. The coordinate does however scale with  $\sqrt{E}$ .

$$e^{-\sqrt{E}(r_1+r_2)} \equiv e^{-\frac{1}{2}\sqrt{E}(z_1+z_2+2z_3)} \quad (4.4.1)$$

$\psi_{\text{cox}}$  refers to the wavefunction used by Cox *et al.*<sup>[73,86,87]</sup> with the exponential form in Eq. (4.4.2). This has a single non-linear variational parameter where the parameter space scales with  $\sqrt{\frac{E}{K}}$ .

$$e^{-\sqrt{\frac{E}{K}}(r_1+r_2)} \equiv e^{-\frac{1}{2}\sqrt{\frac{E}{K}}(z_1+z_2+2z_3)} \quad (4.4.2)$$

$\psi_A$  will refer to the wavefunction with the exponential form in Eq. (4.4.3). This wavefunction constitutes the first of the wavefunctions developed in this work (following the work of Galvez *et al.*<sup>[74,93]</sup>). It has one non-linear variational parameter but does not scale with  $\sqrt{E}$  as the  $\psi_{\text{cox}}$  wavefunction does.  $\psi_A$  and  $\psi_{\text{cox}}$  theoretically cover the same domain, and therefore a  $K$  value, along with the  $\sqrt{E}$  value, can be inter-converted to the  $A$  value in this wavefunction.

$$e^{-\alpha(r_1+r_2)} \equiv e^{-\frac{1}{2}A(z_1+z_2+2z_3)} \quad (4.4.3)$$



#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

$\psi_{AC}$  refers to the wavefunction with the exponential form in Eq. (4.4.4). This is the second wavefunction developed in this work and incorporates freedom for the  $r_3$  variable to appear in the wavefunction. When  $C = 2A$  in Eq. (4.4.4) then  $\gamma = 0$  and this wavefunction takes the same form as in Eq. (4.4.3).

$$e^{-\alpha(r_1+r_2)-\gamma r_3} \equiv e^{-\frac{1}{2}(A(z_1+z_2)+Cz_3)} \quad (4.4.4)$$

$\psi_{ABC}$  will refer to the wavefunction with the exponential form in Eq. (4.4.5). This is the third wavefunction developed in this work and incorporates freedom for all 3 non-linear variational parameters to appear in the exponent with any real positive value. Therefore all 3 inter-particle coordinates are given freedom to appear freely in the asymptotic form. When  $A = B$  in Eq. (4.4.5) then this wavefunction takes the same form as Eq. (4.4.4). Additionally when  $C = A + B$  then  $\gamma = 0$  and when  $A = B$  and  $C = A + B$  then this wavefunction takes the same form as Eq. (4.4.3).

$$e^{-\alpha r_1 - \beta r_2 - \gamma r_3} \equiv e^{-\frac{1}{2}(Az_1 + Bz_2 + Cz_3)} \quad (4.4.5)$$

In the following data, energy convergence is studied by comparing the performance of the different wavefunctions. The first 3 wavefunctions conform to the formal asymptotic solution designed for atomic systems with no  $r_3$  in the asymptote; they perform well for such atomic systems, but as will be shown, perform less well for other systems. The last two wavefunctions were designed for systems not adhering to the formal solution, such as molecular systems. The  $\alpha$  parameter in these wavefunctions according to the formal solution in Eq. (4.2.73) would tend towards  $\sqrt{-2E_{2b}}$  (for an atom with infinite nuclear mass  $\sqrt{-2E_{2b}}$  is  $Z_3$  in Eq. (4.2.76)) and this will be discussed below.

The non-linear variational parameters form an energy surface and it will be shown that for the molecular systems, this surface has more than one minimum. These parameters are optimised using the NROpt or BOBYQA method described earlier (Section 3.2.1 on page 35). All of these systems were calculated with the nucleus in motion, and atomic units, unless otherwise stated. When comparing to literature it is important to keep in mind that the literature values use the 2006 CODATA<sup>[94]</sup> values of recommended physical constants for the masses of particles. However, in this

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

work we use the 2010 CODATA<sup>[18]</sup> values, as a result the values reported here will not necessarily match those reported elsewhere depending on the significant figures that changed between the 2006 and 2010 CODATA values.

**The Atomic  $H^-$  and Molecular  $H_2^+$  Case** The two extremes of 3Body systems namely atomic and molecular are presented first; atomic has a prototypical three-body example in  $H^-$  and molecular has the prototypical three-body example of  $H_2^+$ .  $H^-$  is defined as atomic, as the uniquely charged particle is much heavier whereas  $H_2^+$  is molecular as the two identically charged particles are much heavier than the uniquely charged particle.

**Atomic** The  $H^-$  system is firmly an atomic system with a single, heavy, charged particle. The data in Table 4.4.1 shows that all of the wavefunctions have a comparable energy convergence, and the  $\alpha$  parameter is approximately the same as the charge of the nucleus as described in the formal solution in Eq. (4.2.75). This shows, that at least in these cases, the atomic systems do conform to the  $\Omega_0$  asymptotic region (Section 4.2.7). The He system also firmly conforms to the expected atomic behaviour. It has an  $\alpha$  parameter conforming to the  $\Omega_0$  region and converges rapidly with all types of wavefunction studied here.

The energies at their best have been converged to 11 s.f. for  $H^-$  and 10 s.f. for He. The best literature has this value converged to 33 s.f. and 42 s.f. respectively. Although theoretically the wavefunctions discussed in this work, with sufficient number of terms and numerical precision, could achieve such accuracy, following the discussion by Schwartz,<sup>[62]</sup> such accuracies are deemed unnecessary for real-world application, particularly when considering that these calculations do not include relativistic effects that become important on such high accuracy levels. Additionally the finite masses are only known to 6 s.f. to 12 s.f. and are subject to change upon future revisions of their mass, thereby invalidating the higher accuracy calculations with the older masses. Furthermore the computational time could prove untenable. To understand the level of accuracy these degrees of convergence give, consider that  $1 E_h$  is  $2625.49964 \text{ kJ mol}^{-1}$ <sup>[18]</sup> and that an error on the 10<sup>th</sup> decimal place of  $H^-$  in Hartree represents an error of the order of  $2.6 \times 10^{-7} \text{ kJ mol}^{-1}$  i.e. a tiny error.

4. The Non-Relativistic Ground State Energies of Three-body Systems

H <sup>-</sup>							
N	$\omega$	$\psi_{pekeris}$	K	$\psi_{cor}$	Energy (a.u.)		
					$\alpha$	$\psi_A$	$\alpha, \gamma$
95	8	-0.527 44487841305	1.07	-0.527 44492761832	0.70	-0.527 44492761832	0.77, -0.07
252	12	-0.527 445 83849221	0.84	-0.527 445 85052872	0.79	-0.527 445 85052872	0.84, -0.05
715	18	-0.527 445 87796023	0.63	-0.527 445 88043430	0.92	-0.527 445 88043430	0.94, -0.02
1078	21	-0.527 445 87997099	0.56	-0.527 445 88094272	0.97	-0.527 445 88094272	0.99, -0.02
2856	30	-0.527 445 88101055	0.40	-0.527 445 881 10610	1.15	-0.527 445 881 10610	0.89, 0.27
Literature					-0.527 445 881 109 440 729 069 818 738 420 952 [65]		
He							
N	$\omega$	$\psi_{pekeris}$	K	$\psi_{cor}$	Energy (a.u.)		
					$\alpha$	$\psi_A$	$\alpha, \gamma$
95	8	-2.903 30357004554	0.49	-2.903 30448607774	2.43	-2.903 30448607774	2.12, 0.34
252	12	-2.903 30447113354	0.37	-2.903 304 55573429	2.80	-2.903 304 55573432	2.40, 0.45
715	18	-2.903 304 55150197	0.26	-2.903 304 55769475	3.35	-2.903 304 55769468	3.08, 0.30
1078	21	-2.903 304 55553823	0.23	-2.903 304 557 72202	3.57	-2.903 304 557 72207	2.81, 0.84
2856	30	-2.903 304 55754860	0.15	-2.903 304 557 72943	2.99	-2.903 304 557 72745	3.98, -1.04
Literature					-2.903 304 557 729 580 294 733 816 943 892 697 752 659 27 [65]		

**Table 4.4.1.:** The energy and non-linear variational parameters ( $\alpha, \gamma$  where  $\alpha = \frac{B+C-A}{2}$  and  $\gamma = \frac{A+B-C}{2}$ ) of the atomic systems, H<sup>-</sup> and He, with different number of terms in the wavefunction. The energies are compared to the best available literature values

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

**Molecular** The molecular systems, which have two masses much heavier than the third, behave in a contrasting fashion to the atomic. According to the results as shown in Table 4.4.2, the wavefunctions with the  $r_3$  term in the asymptote ( $\psi_{AC}$  and  $\psi_{ABC}$ ) had the best energy convergence. This is a pronounced difference to that in the atomic case. In the atomic case the  $\gamma$  parameter was much smaller typically less than one. The  $\alpha$  parameter was also larger than expected according to Eq. (4.2.66), as it should also have a value of 1. These results show that the formal solution of the  $\Omega_0$  asymptotic region is not appropriate in the molecular systems.

The results in Table 4.4.2 converged to 9 s.f. this is a great improvement over the  $\psi_{pekeris}$ ,  $\psi_{cox}$  and  $\psi_A$  wavefunctions. Highly accurate works in the literature have converged these systems to 20 s.f..<sup>[95,96]</sup>

**The Systems Between Atomic and Molecular** In the following discussion mass ratio, refers to the ratio of the uniquely charged particle to the identically charged particles. The following systems presented here, have mass ratios between the atomic and molecular extremes in order to investigate the point at which the  $\gamma$  parameter begins to have a significant effect. This is determined by accessing the rate of convergence, and when the  $\alpha$  parameter changes from the formal solution.

**Atom-Like** Table 4.4.3 contains the results of studies on the atom-like “exotic” systems investigated in this chapter. The  $\alpha$  parameter did not conform to the solution of the asymptotic region of  $\Omega_0$ . The  $\gamma$  parameter, although not always zero, had little effect on the convergence rate of the energy. Table 4.4.3 details the results calculated for these atom-like systems.

Atom-like systems are not commonly found in nature.  $\mu\text{H}^-$  is an exotic system in that it has muons instead of electrons. The author is unaware of any experimental data for such a species. However theoretically it has a bound state, as it is a symmetric system.<sup>[97]</sup> The energy value is considerably larger than the other systems and this is because the energy between two oppositely charged particles is proportional to the reduced mass. This energy value matches that reported by Bhattacharyya *et al.*<sup>[63]</sup> The muonium negative ion ( $\text{Mu}^-$ ) was chosen as a lighter isotope of  $\text{H}^-$  to investigate the effect of reducing the mass ratios between the nucleus and the electrons, these are  $\approx \frac{1}{9}$  for  $\mu\text{H}^-$  and  $\approx \frac{1}{206}$  for  $\text{Mu}^-$ .

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

H <sub>2</sub> <sup>+</sup>									
N	ω	ψ <sub>pekeris</sub>	K	ψ <sub>cox</sub>	α	ψ <sub>A</sub>	α, γ	ψ <sub>AC</sub>	
95	8	−0.58465240101414	0.24	−0.59071857801777	0.93	−0.58676230444727	0.63, 4.51	−0.59587731020826	
252	12	−0.58820034208506	0.11	−0.59467249194615	1.68	−0.59432843584602	0.65, 6.04	−0.59701024969794	
715	18	−0.59250534488898	0.07	−0.59664062455832	2.84	−0.59664062455837	0.64, 8.81	−0.59713807694103	
1078	21	−0.59349141200752	0.06	−0.596933460838538	3.11	−0.596933460838546	0.62, 10.72	−0.59713901929802	
2856	30	−0.59531437778790	0.04	−0.59713044394905	3.80	−0.59713044394930	0.55, 13.42	−0.59713906307895	
Literature								−0.59713906312340507483 <sup>[95]</sup>	
D <sub>2</sub> <sup>+</sup>									
No. of Terms	Pekeris Shell (ω)	ψ <sub>pekeris</sub>	K	ψ <sub>cox</sub>	α	ψ <sub>A</sub>	α, γ	ψ <sub>AC</sub>	
95	8	−0.58500803226790	0.24	−0.59123512486365	0.93	−0.58716808714508	0.63, 4.56	−0.59685753314806	
252	12	−0.58862721474605	0.11	−0.59545870695546	2.29	−0.59545870695546	0.78, 3.76	−0.5923778940432	
715	18	−0.59310160805749	0.07	−0.59784302740772	2.87	−0.59784302740771	0.64, 9.06	−0.59877712682031	
1078	21	−0.59414728331407	0.06	−0.59830140432210	3.17	−0.59830140432217	0.71, 8.62	−0.59878704519735	
2856	30	−0.59616809152152	0.04	−0.59874049560776	4.03	−0.59874049560813	0.63, 14.52	−0.59878878407568	
Literature								−0.59878878433068346454 <sup>[95]</sup>	
DT <sup>+</sup>									
No. of Terms	Pekeris Shell (ω)	α	ψ <sub>A</sub>	α, γ	ψ <sub>AC</sub>	α, β, γ	ψ <sub>ABC</sub>		
165	8	0.93	−0.58723584087015	0.63, 4.57	−0.59702700478209	0.63, 0.63, 4.57	−0.59702700494025		
455	12	2.29	−0.59559229489162	0.78, 3.77	−0.59848017313153	0.65, 0.65, 6.15	−0.59870059865736		
1330	18	2.88	−0.59805640873440	0.78, 8.48	−0.59909761663372	0.64, 0.64, 9.10	−0.59911149980841		
2024	21	3.19	−0.59855146188470	0.55, 14.29	−0.59912379110159	0.71, 0.71, 8.66	−0.59912725343495		
5456	30	4.08	−0.59906186912803	0.63, 14.68	−0.59913066184452	0.67, 0.67, 13.45	−0.59913066170882		
Literature								−0.59913066285506263 <sup>[95]</sup>	

**Table 4.4.2.:** The energy and non-linear variational parameters ( $\alpha, \gamma$  where  $\alpha = \frac{B+C-A}{2}$  and  $\gamma = \frac{A+B-C}{2}$ ) of the molecular system with different number of terms in the wavefunction. Compared to best available literature

4. The Non-Relativistic Ground State Energies of Three-body Systems

Mu <sup>-</sup>									
No. of Terms	Pekeris Shell ( $\omega$ )	$\psi_{pekeris}$	K	$\psi_{cox}$	$\alpha$	$\psi_A$	$\alpha, \gamma$	$\psi_{AC}$	
95	8	-0.525 05378220377	1.08	-0.525 05384336179	0.96	-0.525 05384338338	1.08, -0.12	-0.525 05386823994	
252	12	-0.525 05476458309	0.85	-0.525 05477567868	1.09	-0.525 05477567843	1.08, 0.01	-0.525 05477559908	
715	18	-0.525 05480345213	0.67	-0.525 05480579783	1.20	-0.525 05480574078	1.08, 0.01	-0.525 05480487919	
1078	21	-0.525 05480541910	0.67	-0.525 05480628513	1.20	-0.525 05480625797	1.08, 0.01	-0.525 05480593569	
2856	30	-0.525 05480643597	0.40	-0.525 05480652932	1.14	-0.525 05480652937	0.91, 0.23	-0.525 05480652912	
Literature								-0.525 054806223 [98]	
$\mu\text{H}^-$									
No. of Terms	Pekeris Shell ( $\omega$ )	$\psi_{pekeris}$	K	$\psi_{cox}$	$\alpha$	$\psi_A$	$\alpha, \gamma$	$\psi_{AC}$	
95	8	-97.56660411572340	1.27	-97.56675824820690	0.61	-97.56675824820690	0.67, -0.06	-97.56676401646040	
252	12	-97.56698538120730	1.00	-97.56698538125280	0.69	-97.56698538125300	0.73, -0.04	-97.56698543534380	
715	18	-97.56699233717650	0.74	-97.56699255736610	0.80	-97.56699255736650	0.81, -0.01	-97.56699255741010	
1078	21	-97.56699257946550	0.66	-97.56699267823410	0.84	-97.56699267823660	0.85, -0.00	-97.56699267823910	
2856	30	-97.56699270440090	0.47	-97.56699271548160	1.00	-97.56699271548300	0.96, 0.04	-97.56699271548510	
Literature								-97.56698459 [99]	

**Table 4.4.3.:** The energy and non-linear variational parameters ( $\alpha, \gamma$  where  $\alpha = \frac{B+C-A}{2}$  and  $\gamma = \frac{A+B-C}{2}$ ) of the “atom” like system with different number of terms in the wavefunction. Compared to best available literature

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

From Table 4.4.3 the  $\psi_A$  and  $\psi_{AC}$  wavefunctions perform noticeably better than the  $\psi_{pekeris}$  type wavefunction, and slightly better than the  $\psi_{cox}$  wavefunction.

**Molecule-Like** The Positronium negative ion ( $\text{Ps}^-$ ) is an exotic species and, with all particles the same mass, it lies between both atomic and molecular. For reasons detailed later in Section 5.4.1 and in the work by Mátyus *et al.*<sup>[100]</sup> it is considered as a molecule-like species and therefore as a very light isotope of  $\text{H}_2^+$  with a mass ratio of the nucleus to electron of 1:1. All of the wavefunctions performed well, achieving good convergence. Unlike the molecule cases of  $\text{H}_2^+$  etc. the  $\gamma$  in the asymptotic form of wavefunctions is close to zero and behaves like the atomic cases. For some cases however the  $\gamma$  is large for example with  $\text{Mu}_2^+$  here the inclusion of  $\gamma$  benefits the energy conversion by 3 s.f.. The  $\text{d}^+\text{t}^+\mu^-$  system is known to catalyse nuclear fusion at room temperature.<sup>[13,14,64,66,101]</sup> However the inclusion of  $\beta$  parameter resulted in no additional benefit to the convergence of the energy. This is of no surprise as the mass ratios of d to t relative to d to  $\mu$  are much smaller, leading to an anticipated lack or deviation between the  $\alpha$  and  $\beta$  parameters of this system, resulting in no additional benefit to the rate of convergence. In Table 4.4.5 an example is given of a system where  $\beta$  is important to the rate of convergence. This is a system of importance in Chapter 5. In that chapter many systems with significantly different  $\alpha$  and  $\beta$  were determined. The details of these systems and why they were calculated are given in Chapter 5.

Mu <sub>2</sub> <sup>+</sup>							
No. of Terms	Pekeris Shell ( $\omega$ )	$\psi_{pekeris}$	K	$\psi_{cor}$	Energy (a.u.)		
					$\alpha$	$\alpha, \gamma$	$\psi_{AC}$
95	8	-0.57922916010889	0.25	-0.58330461380319	1.52	0.71, 3.00	-0.58507760233610
252	12	-0.58183136872110	0.13	-0.58489116245487	2.10	0.65, 5.09	-0.58512605342867
715	18	-0.58423913972842	0.10	-0.58512172121133	2.41	0.74, 6.14	-0.58512609825197
1078	21	-0.58463483021853	0.10	-0.58512561989213	2.48	0.68, 4.66	-0.58512609825234
2856	30	-0.58504435072678	0.08	-0.58512609780042	2.69	0.66, 9.37	-0.58512609825234
Literature							-0.585126097219193 [74]
$\mu\text{H}_2^+$							
No. of Terms	Pekeris Shell ( $\omega$ )	$\psi_{pekeris}$	K	$\psi_{cor}$	Energy ( $\mu\text{a.u.}$ )		
					$\alpha$	$\alpha, \gamma$	$\psi_{AC}$
95	8	-0.49437281754085	0.43	-0.49438532677260	1.07	0.63, 0.70	-0.49438675428741
252	12	-0.49438604822256	0.38	-0.49438680896421	1.14	0.78, 0.59	-0.49438681215287
715	18	-0.49438680955018	0.32	-0.49438681227666	1.23	1.00, 0.56	-0.49438681227866
1078	21	-0.49438681202476	0.27	-0.49438681227889	1.36	1.08, 0.62	-0.49438681227926
2856	30	-0.49438681227464	0.19	-0.49438681227938	1.25	0.85, 0.45	-0.49438681227930
Literature							-0.494386815212835026522266 [8]
Ps <sup>-</sup>							
No. of Terms	Pekeris Shell ( $\omega$ )	$\psi_{pekeris}$	K	$\psi_{cor}$	Energy (a.u.)		
					$\alpha$	$\alpha, \gamma$	$\psi_{AC}$
95	8	-0.26199394389281	2.17	-0.26200460964796	0.35	0.36, -0.01	-0.26200467074623
252	12	-0.26200487647636	1.68	-0.26200505940751	0.40	0.39, 0.00	-0.26200505940958
715	18	-0.26200506371228	1.20	-0.2620050700740	0.47	0.43, 0.04	-0.26200507001072
1078	21	-0.26200507017933	1.05	-0.26200507018297	0.50	0.48, 0.02	-0.26200507018305
2856	30	-0.26200507023024	0.75	-0.26200507023188	0.59	0.29, 0.22	-0.26200507022755
Literature							-0.2620050702329801077 [102]
$\mu\text{DT}^+$							
No. of Terms	Pekeris Shell ( $\omega$ )	$\alpha$	$\psi_A$	$\alpha, \gamma$	Energy ( $\mu\text{a.u.}$ )		
					$\psi_{AC}$	$\alpha, \beta, \gamma$	$\psi_{ABC}$
165	8	1.28	-0.53857521178684	0.65, 1.45	-0.53859489913406	0.61, 0.69, 1.46	-0.53859490504787
455	12	1.38	-0.53859485817061	0.80, 1.33	-0.53859497048322	0.76, 0.86, 1.33	-0.53859497048322
1330	18	1.38	-0.53859497047080	1.05, 1.11	-0.53859497052712	1.18, 0.79, 1.00	-0.53859497052709
2024	21	1.53	-0.53859497052613	1.27, 0.75	-0.53859497052717	0.68, 0.95, 1.75	-0.53859497052715
5456	30	1.13	-0.53859497052710	0.81, 0.42	-0.53859497052717	0.75, 0.82, 0.35	-0.53859497052711
Literature							-0.538594971709480730 [8]

**Table 4.4.4.:** The energy and non-linear variational parameters ( $\alpha, \gamma$  where  $\alpha = \frac{B+C-A}{2}$  and  $\gamma = \frac{A+B-C}{2}$ ) of the “molecule” like systems with different number of terms in the wavefunction.  $\mu\text{a.u.}$  is muon atomic units where the mass of the muon is unity rather than the electron. Compared to best available literature



$p^+m_{cr} + e^-$		Energy (a.u.)					
No. of Terms	Pekeris Shell ( $\omega$ )	$\alpha$	$\psi_A$	$\alpha, \gamma$	$\psi_{AC}$	$\alpha, \beta, \gamma$	$\psi_{ABC}$
165	8	0.47	-0.49883031633113	0.50, -0.03	-0.49883149740972	1.00, -0.00, 0.00	-0.49972783971223
455	12	0.37	-0.49946875189855	0.50, -0.13	-0.49947269637679	1.03, 0.30, -0.30	-0.49972783971222
1330	18	0.29	-0.49965930368076	0.50, -0.22	-0.49966159498886	1.02, 0.34, -0.34	-0.49972783971223
2024	21	0.26	-0.49968789595638	0.50, -0.24	-0.49968951402582	1.03, 0.45, -0.45	-0.49972783971223

**Table 4.4.5.:** A system where the  $\beta$  parameter is important for the convergence of the energy. This system uses the concept  $m_{cr}$  that will be described later in Chapter 5 in this case this mass is  $\approx 2.21$  a.u.

### 4.4.2. The Energy Surfaces of the Non-Linear Parameters

In this section the effects that the non-linear variational parameters have on the energy is shown in a graphical form. These energy surfaces are displayed as contour plots. The contour plots provide insight into how the parameters affect the energy of the wavefunction, revealing patterns and general trends that can be used to best decide how to find global minima. In the following contours the  $Z$ -axis is the same used in Figures 3.2.1 and 3.2.2 (which is  $\text{Log}_{10}(E(A, C) - E_{\min})$ ). This scales the contours such that topological detail can be seen regardless of the apparent flatness of the surface. The important aspect to this discussion is the different topologies and not the absolute values of the  $Z$ -axis and therefore absolute contour values are not shown.

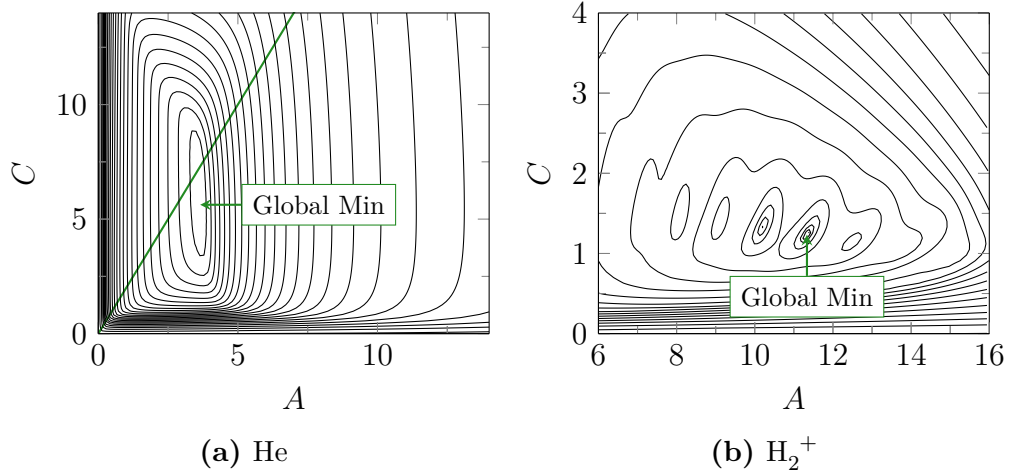
**Atom and Molecule Energy Surfaces** It is seen that for atomic systems the energy surfaces have a single minimum along the  $C = 2A$  line, where  $\gamma = 0$  (this line is shown in Figure 4.4.1a as a green line). In molecular systems this is a more complex case with multiple minima away from the  $C = 2A$  line where  $\gamma \neq 0$ .

**Atom: He** Figure 4.4.1a is the contour plot of the  $A$  and  $C$  non-linear variational parameters against energy on the  $x$   $y$  and  $z$  axis respectively of helium. There is only a single minimum on the surface. This minimum is found when  $C = 2A$  that is when  $\gamma = 0$ . This energy surface is large and flat and continues to get flatter as the number of terms in the wavefunction increases, as is shown in Section 4.4.2.

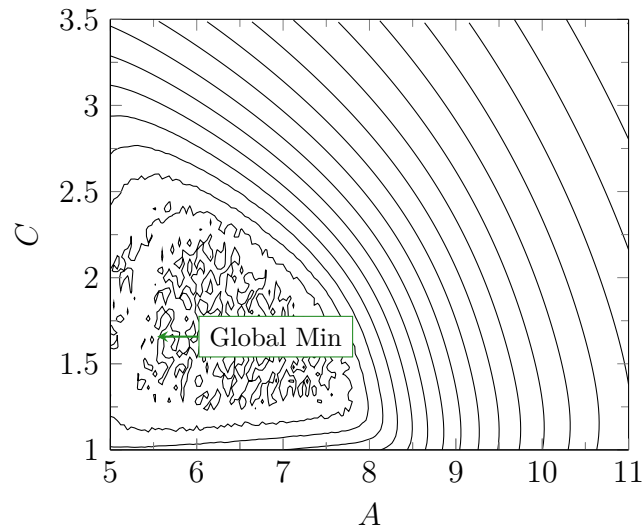
**Molecule:  $\text{H}_2^+$**  The molecular hydrogen cation is used as the prototypical example of a molecule. Figure 4.4.1b shows that the energy surface is complex in contrast to the He case (Figure 4.4.1a). Multiple minima are present over a range of values. The lowest minimum is found to the right of the area where  $C = 2A$  and  $\gamma = 0$ . This was found to also be the case for all molecular systems investigated. It became necessary to take extra attention to minimise these values to their global minimum. This minimisation was done using the steps outlined in Section 3.4.1 on page 76.

**Exotic Systems** The contour plots in the region between archetypal atoms and molecules shows a progression from a single minimum; to multiple minima clustered

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems



**Figure 4.4.1.:** The  $A$   $C$  contour plots of the He (a) and  $H_2^+$  (b) system, with a 1078-term wavefunction.



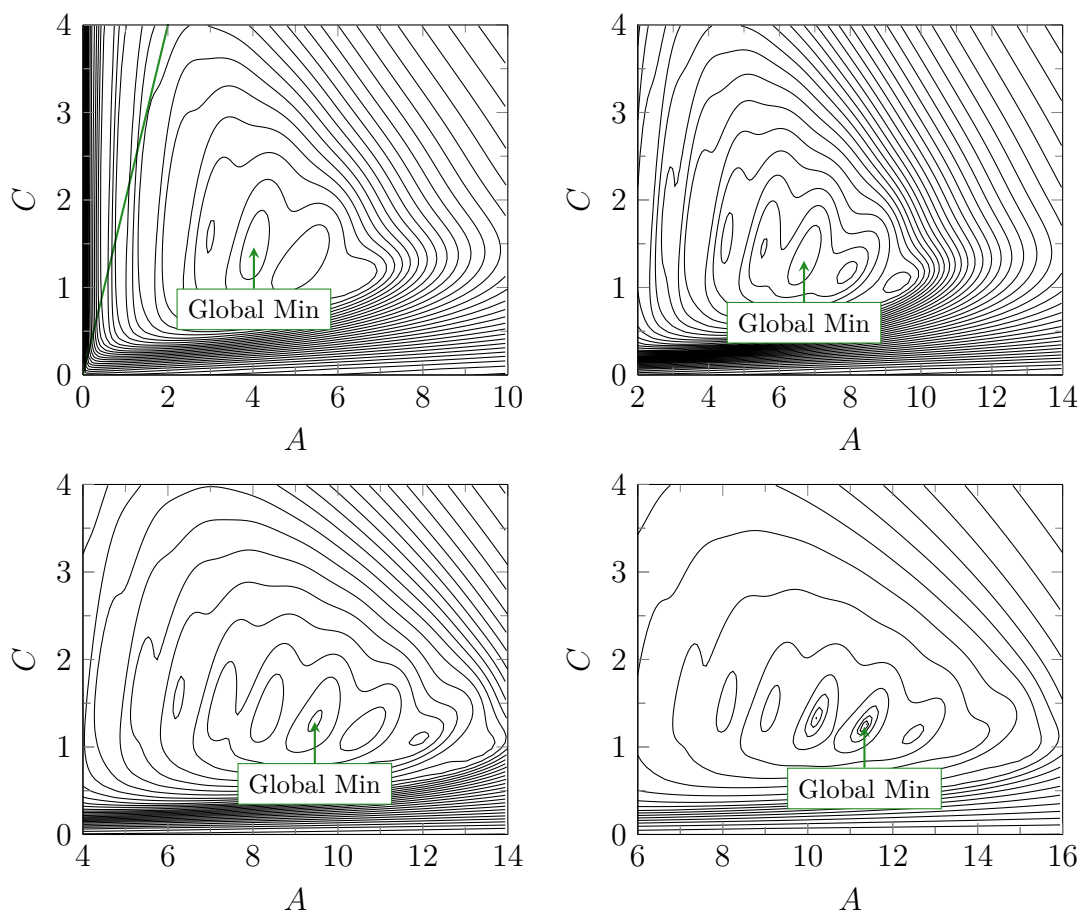
**Figure 4.4.2.:** The  $A$   $C$  contour plots of the  $\mu H_2^+$  system, with a 1078-term wavefunction.

together; to multiple minima spread out over a wide domain.

**Molecule-Like:  $\mu H_2^+$**  The molecule-like “exotic” case of the energy surface was chosen as  $\mu H_2^+$ . Figure 4.4.2 is the 1078  $A$   $C$  contour plot of the energy. It behaves similar to the molecule case seen in Figure 4.4.1b however this case has fewer minima that are more closely packed towards the to origin.

**The Effects of Number of Terms in the Wavefunction** With the atom and atom-like cases, increasing the number of terms in the wavefunction causes the  $A$  and  $C$  to change only subtly. However in the molecule and molecule-like cases, a dramatic

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems



**Figure 4.4.3.:** The  $A$   $C$  contour plots of the  $H_2^+$  system, with a different numbers of term wavefunction.

change could be observed. The non-linear variational parameters would shift further away from the  $C = 2A$  case as the number of terms increased. Figure 4.4.3 show this for the cases of  $H_2^+$ . For example with a 95-term wavefunction the minimum for  $H_2^+$  is located at  $A \approx 6.5$ ,  $C \approx 1.5$ , increasing the number of terms to 715 terms shifted the  $A$  parameter to  $\approx 9.5$  whilst  $C$  remained the same and increasing the number of terms again to 1078 terms shifted  $A$  to approximately 15. Exotic molecule-like systems such as  $Mu^-$  also have this shifting minima effect but with a slower, less pronounced shift as the number of terms increased.

#### 4.4.3. Quality of the Wavefunction and Expectation Values

In addition to the energy of a system, wavefunctions can be used to calculate a variety of other properties. Operators such as the Hamiltonian are mathematical constructs that can represent physical observables. Any two operators that commute

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

have a shared eigenstate or wavefunction. As a result of this, for any operator that commutes with the Hamiltonian we can use the wavefunction calculated previously to calculate the expectation value of the associated observable. These operators that represent physical observables are self-adjoint and bounded.<sup>[103,104]</sup> Here the quality of the  $\psi_{AC}$  and  $\psi_{ABC}$  wavefunctions are investigated. These two wavefunctions are looked into with greater detail because they are capable of calculating both atomic and molecular systems alike. They are the most parametrised and therefore may be sacrificing quality of the wavefunction for a lower energy.

**Expectation Values** To calculate observables, expectation values are used. The resulting expectation value represents the average value of the observable should it be measured with no errors an infinite number of times. This makes expectation values an excellent measure, since they give the value experimentalists would, on average, observe.

Expectation values are calculated with the following equation where  $\hat{\Omega}$  represents any operator of an observable and  $\omega$  the measurable value of that observable.  $\langle \hat{\Omega} \rangle$  is the shorthand notation for the expectation value of that operator:

$$\langle \hat{\Omega} \rangle = \frac{\int \psi^* \hat{\Omega} \psi d\tau}{\int \psi^* \psi d\tau} = \omega \quad (4.4.6)$$

Having a wavefunction that produces good energy values does not necessarily translate to a good wavefunction that can calculate accurately any expectation value. There are a few indications that can be used to test the quality of the wavefunction where a quality wavefunction is defined as giving good expectation values. The virial condition is one of these tests.

**The Virial Condition** is a property of a bound state system that can be calculated for any system with a kinetic and potential energy. The virial condition states that the potential energy divided by the kinetic energy will have the following relationship:

$$\frac{\langle \hat{V} \rangle}{\langle \hat{T} \rangle} = \frac{2}{\nu} \quad (4.4.7)$$

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

System	Virial Value	Virial Value +2
H <sup>-</sup>	-2.000 000 000 005 35	$-5.35 \times 10^{-12}$
He	-2.000 000 000 006 13	$-6.13 \times 10^{-12}$
H <sub>2</sub> <sup>+</sup>	-1.999 999 999 397 95	$6.02 \times 10^{-10}$
D <sub>2</sub> <sup>+</sup>	-2.000 000 000 192 07	$-1.92 \times 10^{-10}$
DT <sup>+</sup>	-1.999 999 999 999 83	$1.71 \times 10^{-13}$
Mu <sup>-</sup>	-2.000 000 000 003 98	$-3.98 \times 10^{-12}$
$\mu$ H <sup>-</sup>	-2.000 000 000 003 10	$-3.10 \times 10^{-12}$
Mu <sub>2</sub> <sup>+</sup>	-1.999 999 999 997 09	$2.91 \times 10^{-12}$
$\mu$ H <sub>2</sub> <sup>+</sup>	-2.000 000 000 001 09	$-1.09 \times 10^{-12}$
Ps <sup>-</sup>	-2.000 000 000 004 22	$-4.22 \times 10^{-12}$
$\mu$ DT <sup>+</sup>	-1.999 999 999 999 83	$1.71 \times 10^{-13}$

**Table 4.4.6.:** The virial condition of various systems. A virial condition of  $-2$  indicates a good quality wavefunction. The last column (Virial Value +2) ( $\langle\eta\rangle$ ) demonstrates difference between the virial condition calculated here and the true virial value.

For a general potential of the form:

$$\hat{V} = \sum_{i,j} a_{ij} r_{ij}^{\nu} \quad (4.4.8)$$

For the coulomb potential well,  $\nu = -1$ . This means that for the systems investigated here:

$$\frac{\langle\hat{V}\rangle}{\langle\hat{T}\rangle} = -2 \quad (4.4.9)$$

Table 4.4.6 reveals the results of this test, which indicate that good quality wavefunctions are calculated and that accurate expectation values can be calculated in principle.

**Cusp Conditions** In addition to the virial condition the cusp condition is another measure of wavefunction quality employed in this work. The cusp is the gradient of the wavefunction at the point of coalescence of two or more particles. In the inter-particle coordinate system ( $r_i$ ) this occurs when  $r_i = 0$ . This gradient has a known calculable value. As the wavefunction approaches zero the potential energy becomes singular<sup>[105]</sup> (as  $\frac{Z_i Z_j}{r_k}$ ), however it still must remain self-adjoint and bounded.<sup>[105]</sup> Therefore the kinetic energy must compensate. Kato<sup>[105]</sup> used theorems based on a

#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

self-adjoint and bounded Hamiltonian to form a relationship between the potential and the kinetic energy at this singularity. For further details, refer to the work by Kato.<sup>[105]</sup> The cusp values have the notation  $\nu_{ij}$  where  $i$  and  $j$  denote that this is the cusp where particles  $i$  and  $j$  coalesce. The exact cusp conditions can be calculated with Eq. (4.4.10).<sup>[105]</sup>

$$\nu_{ij} = Z_i Z_j \frac{m_i m_j}{m_i + m_j} \quad (4.4.10)$$

Where  $m_i$  and  $m_j$  are the masses of particles  $i$  and  $j$  and where  $Z_i$  and  $Z_j$  are the charges. Kato<sup>[105]</sup> first derived the cusp condition (with a spherical average restriction) to clarify some of the singularities of the He wavefunction. Bingel<sup>[106]</sup> later refined this by integrating Kato's condition and removing the spherical average restriction. The cusps are calculated in two different ways. The expansion method is one proposed by Pack and Byers Brown.<sup>[57]</sup> These authors extended the method to finite mass systems where all particles are in motion and proposed an expression of Bingel's form of the cusp in the form of an expansion. The second method uses Dirac deltas, from the generic description:

$$\nu_{ij} = \left. \frac{\partial \psi}{\partial r_{ij}} \right|_{r_{ij}=0} \quad (4.4.11)$$

Expressing this with Dirac deltas gives Eq. (4.4.12). (A detailed discussion of Dirac deltas is given in Section 6.2.2 on page 173)

$$\nu_{ij} = \frac{\int_0^\infty \psi^* \frac{\partial}{\partial r_{ij}} \psi \delta(r_{ij}) d\tau}{\int_0^\infty \psi^* \psi \delta(r_{ij}) d\tau} \quad (4.4.12)$$

Table 4.4.7 shows the calculated cusp values for both methods. For each system the exact cusp (from Eq. (4.4.10)) is given in the final column.

From Table 4.4.7 it can be seen that for atomic systems the cusp condition is well defined and equates to the exact value to several significant figures. For molecular systems however, the wavefunction fails to calculate the exact nucleus-nucleus cusp with either method. The Dirac method is the more accurate of the two giving values closer to the exact value. The cusps for the molecule systems and even the molecule-like systems are difficult properties to calculate.<sup>[107]</sup> To the authors knowledge, no accurate calculation of the cusps for such systems have been made. Frolov *et al.*

System	Expansion Cusps		Dirac Cusps		Exact Cusps	
	$\nu_{31}$	$\nu_{21}$	$\nu_{31}$	$\nu_{21}$	$\nu_{31}$	$\nu_{21}$
$H^-$	-0.989 588 960	0.486 430 258	-0.999 454 076	0.499 877 401	-0.999 455 679	0.500 000 000
He	-1.994 257 88	0.489 774 748	-1.999 721 44	0.499 939 037	-1.999 725 85	0.500 000 000
$H_2^+$	2.539 226 61	-624.708 749	-0.999 457 309	-13.513 853 7	-0.999 455 679	918.076 336
$D_2^+$	-0.606 138 634	-495.918 723	-0.999 728 376	-14.672 935 4	-0.999 727 630	1835.241 48
$Mu^-$	-0.985 523 714	0.486 351 546	-0.995 185 535	0.499 878 576	-0.995 186 946	0.500 000 000
$\mu H^-$	-0.889 830 491	0.484 182 707	-0.898 788 249	0.499 877 576	-0.898 787 918	0.500 000 000
$Mu_2^+$	-1.805 576 32	-271.183 693	-0.995 186 991	-9.372 857 36	-0.995 186 946	103.384 142
$\mu H_2^+$	-0.894 583 963	4.388 836 89	-0.898 787 993	4.439 833 19	-0.898 787 918	4.440 121 65
$Ps^-$	-0.494 342 116	0.483 101 963	-0.500 000 798	0.499 929 349	-0.500 000 000	0.500 000 000

**Table 4.4.7.:** The cusp conditions for various three-body systems using the 2856  $AC$  wavefunction. Using both the expansion method and Dirac delta method.



#### 4. The Non-Relativistic Ground State Energies of Three-body Systems

discuss this difficulty and its implications in calculating related properties, such as nuclear fusion.<sup>[107]</sup> To understand the reasons for inaccurate cusp values with molecular systems but not atomic, the probability densities at the origin of the coordinate are considered. The gradient of the wavefunction at the origin defines the cusp and therefore a well defined probability density at the origin, theoretically, will have a well defined cusp. The probability of finding the two like-charge particles (protons) near each other in  $\text{H}_2^+$  and other molecular systems is low compared to atomic systems (in this latter case the probability of two electrons being near each other in a singlet state).<sup>[74,108]</sup> Therefore an approximate wavefunction need only be accurate in the areas away from the origin to obtain a good energy. For the area at the origin, the trial wavefunction can simply set this probability density to zero, as for all intents and purposes it is zero to numerical precision. Frolov<sup>[109]</sup> shows that the probability of coalescence is too high in current calculations as these values would result in a much easier nuclear fusion than is found in reality. Conversely for atomic ground state  $^1\text{S}$  systems, the probability density is located at the origin and therefore the wavefunction needs to describe this region accurately to determine the energy accurately. To conclude, the cusp indicates the quality of the wavefunction at the origin. For atomic systems we describe this origin accurately. For molecular systems however this region is not described accurately here or anywhere else. However as can be seen from the probability densities<sup>[74,108]</sup> the origin is a relatively unimportant region for molecular systems, except for nuclear fusion and other processes that are based on coalescence of these particles.

**Expectation Values** A good quality wavefunction is defined here as being able to calculate accurate expectation values. Therefore the accuracy of various expectation values calculated here, against available literature, provide a direct measure of the quality of the wavefunction. Table 4.4.8-a shows the calculated expectation values compared to available literature. These expectation values are for a number of inter-particle distances and for the probability of the particles coalescing,  $\langle\delta(r_1)\rangle$  for the probability of particles 1 and 3 coalescing and  $\langle\delta(r_3)\rangle$  for particles 1 and 2.  $\langle\delta(r_2)\rangle$  is not included for symmetric systems as  $\langle\delta(r_1)\rangle \equiv \langle\delta(r_2)\rangle$  where  $m_1^{Z_1} = m_2^{Z_2}$ .

Tables 4.4.8-a and 4.4.8-b shows that the values calculated here are in excellent agreement with available literature, over the complete range from atomic systems to

Expectation	H <sup>-</sup>	He	H <sub>2</sub> <sup>+</sup>	D <sub>2</sub> <sup>+</sup>
$\langle r_1 \rangle$	<b>2.712 095 62</b> 208295 2.712 095 626 604 19, <sup>[110]</sup>	<b>0.929 607 915 01</b> 453 0.929 607 915 012 839, <sup>[111]</sup>	<b>1.692 966 20</b> 855946 1.692 966 209, <sup>[107]</sup>	<b>1.682 346 5</b> 4810708 1.682 346 57, <sup>[107]</sup>
$\langle r_1^2 \rangle$	<b>11.931 747</b> 55276340 11.931 747 756 277 6, <sup>[110]</sup>	<b>1.193 834 894 42</b> 987 1.193 834 894 422 79, <sup>[111]</sup>	<b>3.558 797 931</b> 21576 3.558 797 931, <sup>[107]</sup>	<b>3.507 52</b> 799307172 3.507 528 085, <sup>[107]</sup>
$\langle r_1^{-1} \rangle$	<b>0.682 853 384 8</b> 7907 0.682 853 384 851 18, <sup>[110]</sup>	<b>1.688 076 584 66</b> 406 1.688 076 584 669 63, <sup>[111]</sup>	<b>0.842 492 962 3</b> 2598 0.842 492 962 3, <sup>[107]</sup>	<b>0.845 615</b> 40807392 0.845 615 399 5, <sup>[107]</sup>
$\langle r_3 \rangle$	<b>4.415 692</b> 59457242 4.415 692 603 55, <sup>[65]</sup>	<b>1.422 247 512 60</b> 988 1.422 247 512 6, <sup>[65]</sup>	<b>2.063 913 86</b> 747430 2.063 913 868, <sup>[107]</sup>	<b>2.044 070 0</b> 4710094 2.044 070 089, <sup>[107]</sup>
$\langle r_3^2 \rangle$	<b>25.237 175</b> 20010980 25.237 175 606 801 4, <sup>[110]</sup>	<b>2.517 061 842 4</b> 9890 2.517 061 842 482 15, <sup>[111]</sup>	<b>4.313 285 94</b> 425604 4.313 285 946, <sup>[107]</sup>	<b>4.215 643</b> 12750189 4.215 643 294, <sup>[107]</sup>
$\langle r_3^{-1} \rangle$	<b>0.310 815 007</b> 54935 0.310 815 007 474 003, <sup>[110]</sup>	<b>0.945 697 223 2</b> 2051 0.945 697 223 219 419, <sup>[111]</sup>	<b>0.490 707 798 50</b> 279 0.490 707 798 5, <sup>[107]</sup>	<b>0.493 653 2</b> 4811125 0.493 653 239 5, <sup>[107]</sup>
$\langle \delta(r_1) \rangle$	<b>0.164 279 9</b> 2893023 0.164 279 944 1, <sup>[110]</sup>	<b>1.809 672</b> 26053119 1.809 672 434 886 57, <sup>[111]</sup>	<b>0.206 736</b> 51098839 0.206 736 288, <sup>[107]</sup>	<b>0.207 727</b> 56570405 0.207 727 169, <sup>[107]</sup>
$\langle \delta(r_3) \rangle$	<b>0.002 731 5</b> 9692962 0.002 731 572 79, <sup>[110]</sup>	<b>0.106</b> 30013795116 0.106 299 956 712, <sup>[111]</sup>	<b>0.000 000 000</b> 08624 0.000 000 000 624 1, <sup>[107]</sup>	<b>0.000 000</b> 12446228 0.000 000 020 5, <sup>[107]</sup>

**Table 4.4.8-a.:** The expectation values for various three-body systems compared against literature values. The lower values of each row represent the literature values. Digits in bold match those in the literature.

Expectation	$\text{Mu}^-$	$\text{Mu}_2^+$	$\mu\text{H}_2^+$	$\text{Ps}^-$
$\langle r_1 \rangle$	<b>2.727 182 9</b> 7599017 2.727 182 982 491 54, <sup>[112]</sup>	<b>1.769 302 444</b> 88720 1.769 302 444, <sup>[107]</sup>	<b>2.385 666</b> 64674173 2.385 666 585 6, <sup>[107]</sup>	<b>5.489 633 2</b> 4694317 5.489 633 252 359 44, <sup>[113]</sup>
$\langle r_1^2 \rangle$	<b>12.074 193</b> 75556250 12.074 193 983 236 3, <sup>[112]</sup>	<b>3.938 458 0</b> 2114852 3.938 458, <sup>[107]</sup>	<b>7.769 50</b> 426431325 7.769 503 814 4, <sup>[107]</sup>	<b>48.418 93</b> 675860530 48.418 937 226 238 5, <sup>[113]</sup>
$\langle r_1^{-1} \rangle$	<b>0.679 654 50</b> 115325 0.679 654 500 731 274, <sup>[112]</sup>	<b>0.820 339 76</b> 123123 0.820 339 760 9, <sup>[107]</sup>	<b>0.670 302 5</b> 6600959 0.670 302 578 37, <sup>[107]</sup>	<b>0.339 821 023 0</b> 6886 0.339 821 023 059 22, <sup>[113]</sup>
$\langle r_3 \rangle$	<b>4.439 280 0</b> 7993060 4.439 280 092 007 01, <sup>[112]</sup>	<b>2.205 215 23</b> 290632 2.205 215 237, <sup>[107]</sup>	<b>3.299 486</b> 29061553 3.299 486 184 357 38, <sup>[10]</sup>	<b>8.548 580 6</b> 4436396 8.548 580 655 099 18, <sup>[113]</sup>
$\langle r_3^2 \rangle$	<b>25.514 53</b> 590822630 25.514 536 362 257, <sup>[112]</sup>	<b>5.036 585 5</b> 3554045 5.036 585 56, <sup>[107]</sup>	<b>12.390 40</b> 933400740 12.390 408 464, <sup>[107]</sup>	<b>93.178 63</b> 291535270 93.178 633 847 981 1, <sup>[113]</sup>
$\langle r_3^{-1} \rangle$	<b>0.309 199 38</b> 925048 0.309 199 388 975 496, <sup>[112]</sup>	<b>0.470 427 325</b> 95619 0.470 427 325 5, <sup>[107]</sup>	<b>0.351 831 5</b> 0746126 0.351 831 516 25, <sup>[107]</sup>	<b>0.155 631 905 6</b> 7509 0.155 631 905 652 48, <sup>[113]</sup>
$\langle \delta(r_1) \rangle$	<b>0.162 150 6</b> 6879845 0.162 150 681 5, <sup>[112]</sup>	<b>0.198 930</b> 50210935 0.198 930 331, <sup>[107]</sup>	<b>0.131 500 8</b> 5616580 0.131 500 861 437 8, <sup>[8]</sup>	<b>0.020 733 19</b> 995659 0.020 733 198 005 165, <sup>[113]</sup>
$\langle \delta(r_3) \rangle$	<b>0.002 681</b> 70417321 0.002 681 680 5, <sup>[112]</sup>	<b>0.000 000 000</b> 00080 0.000 000 000 270 6, <sup>[107]</sup>	<b>0.000 039 370</b> 67360 0.000 039 370 034 773, <sup>[8]</sup>	<b>0.000 170 99</b> 853526 0.000 170 996 756 79, <sup>[113]</sup>

**Table 4.4.8-b.:** The expectation values for various three-body systems compared against literature values. The lower values of each row represent the literature values. Digits in bold match those in the literature.

molecular systems.

**Quality of the Wavefunction** The quality of the wavefunction from the 3 measures we have given here, suggests that the wavefunction is of sufficient quality to calculate expectation values and thus real-world quantities other than the energy. This means that neither the  $\psi_{AC}$  or the  $\psi_{ABC}$  wavefunctions sacrifice quality of the wavefunction for a lower energy as far as can be determined and therefore both wavefunctions are of sufficient quality for further use throughout this work.

## 4.5. Summary & Conclusions

In summary, the wavefunction used here unifies the treatment of molecular and atomic systems with a single wavefunction that can accurately model both atomic systems, molecular systems and, in addition, be used for the exotic systems in between. This is done with the time independent Schrödinger equation with all particles kept in motion in a translation-free internal coordinate representation. It was found that the inclusion of a single non-linear variational parameter does help the convergence of atomic species compared to the original work on Laguerre polynomial wavefunctions by Pekeris.<sup>[69–71]</sup> It does not however benefit systems where two particles are much heavier than the third, such as in  $\text{H}_2^+$ . For these this work found that the inclusion of a second non-linear variational parameter  $\gamma$  results in a significant increase in the rate of convergence of the energy, at most to be an additional 7 s.f. better convergence. The third parameter  $\beta$  was shown to be of importance for systems with significant difference between particle 1 and particle 2 (the like-charged particles). When there is little mass difference between these two particles then the  $\beta$  parameter did not help the rate of energy convergence. Based on the work of Drake *et al.*<sup>[12,62]</sup> however this  $\beta$  parameter may prove useful for excited states.

The use of these wavefunctions has allowed us to calculate excellent energies and wavefunctions for a variety of systems. The quality of the wavefunction has been confirmed with three different measures. It was found that atomic systems have a well-defined wavefunction over all regions tested including at coalescence. For molecular systems there was some difficulty calculating the wavefunction at the

#### 4. *The Non-Relativistic Ground State Energies of Three-body Systems*

like charge particle coalescence accurately enough for the evaluation of the cusps. However it was concluded that this region is of much less importance for molecular systems as there is a low probability of finding the two heavy particles at coalescence.

Therefore the wavefunction used here is of appropriate quality to calculate other properties and phenomena for three-body systems. In order to treat both atomic and molecular systems equally throughout the  $\psi_{AC}$  wavefunction is used herein for symmetric systems (where particle 1 equals particles 2) and the  $\psi_{ABC}$  is used for asymmetric systems (where particle 1 is not equal to particles 2).

# 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

## 5.1. Introduction

Several investigators have attempted to predict the stability of coulombic three-body systems based on their mass ratios.<sup>[28,97,114–122]</sup> Whereas some have attempted to use mass ratios to predict the stability of specific exotic systems such as  $p^+e^+e^-$ <sup>[116,117]</sup> and  $p^+\mu^-e^-$ <sup>[114,121]</sup> others have focussed on developing a general understanding of the role that mass ratios play in the binding energy of coulombic three-particle systems.<sup>[28,97,117–119]</sup> Stability of a system in this work is defined as the existence of at least one bound state below the lowest continuum threshold.<sup>[28]</sup>

Here, three-body systems of the form  $\{m_1^\pm m_2^\pm m_3^\mp\}$  are considered interacting via a coulomb potential but without recourse to the clamped-nucleus or BO approximation. In the case of three-body atoms or ions, there is a unique choice of a body fixed coordinate system for states with angular momentum  $J = 0$ , namely, that in which the three particles define a plane and the translation-free internal coordinates are chosen to be the inter-particle distances.<sup>[73]</sup> (For details see Section 4.2 on page 89)

The helium atom and hydrogen molecule and their ions have been the “laboratory” of quantum chemistry for over 80 years, driving attempts to accurately model the correlated motion of electrons and the chemical bond.

The exact minimum mass required to bind a, unit-charged, particle to a hydrogen atom was debated during the late 60’s; this mass will henceforth be called the critical mass ( $m_{cr}$ ). Of particular interest was the possibility of binding a positron

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

to a hydrogen to form the  $p^+e^+e^-$  system. Early calculations could not confirm this system as unstable,<sup>[123]</sup> in the early 70's however it was proved as unstable.<sup>[124]</sup> Recently computer resources and techniques have become available to calculate the critical mass, which is the point between stable and unstable, to a much more reliable value.<sup>[120]</sup> As shown in Chapter 4 we have a suitable technique for calculating accurate energies and wavefunctions, for a variety of systems ranging over both atomic and molecular types. We are therefore in a prime position to calculate the minimum mass that will form a bound three-body species. A stability boundary, or zone, is calculated in which all systems within this zone are deemed stable. This looks at stability of three-body systems in terms of mass as a whole, detailing whole ranges of masses as stable. Such zones have been calculated and refined over the years by various authors. Poshusta's work in 1985<sup>[117]</sup> forms a stability zone using perturbation theory under the infinite mass approximation. It shows an early zone of stability, revealing that certain mass ranges can be deemed stable rather than individual systems. However it is limited by the BO approximation and Poshusta concludes that the BO method is not adequate to describe critical masses, but rather provides interpretations for a generalised stability zone. In this work we use our accurate methodology, and two different methods, to calculate the stability zone. The comparison between these methods gives greater insight into the importance of various factors on the stability of three-body systems. The first method labelled as Method 1 (Section 5.2.5), was developed by Martin *et al.*<sup>[97]</sup> who in their 1992 paper used this method to approximate the stability boundaries. In this chapter, as in our recent paper<sup>[125]</sup> we used modern wavefunctions and techniques to recalculate these boundaries more accurately, before comparing to Method 2 (Section 5.2.6). Rebane *et al.*<sup>[119,120,126–129]</sup> is the primary author of what will later be referred to as Method 2. In this work however we use a different wavefunction, calculate critical masses for different systems and provide different analysis to the data. Rebane calculates critical masses using a flexible wavefunction with 60 non-linear variational parameters, for three-body systems involving particles such as protons, deuterons, tritons, muons, kaons and pions. These critical masses are the best currently available in the literature and are used for comparison where possible. In addition to these two methods one could also directly calculate the energy of a three-body

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

system and also calculate its associated lowest continuum threshold energy then change the mass of a particle until the system becomes unstable such that these two energies are equal. Although this is an inefficient method it is perhaps the most conceptually simple, and we label this Method 0 (Section 5.2.2). In 1992 Frolov<sup>[118]</sup> calculated critical masses by varying the masses of the particles until they became unstable using Method 0. Frolov used this method for 8 systems, most of which do not exist and were chosen to best map out the stability zone in general. The results were plotted on a normalised mass, ternary diagram. The ternary diagrams used in this chapter are based on a different set of coordinate involving normalised reciprocal masses as described in Section 5.2.1.

The critical mass also presents an intriguing problem for theoreticians to calculate. Such a system lies on the border between stable and unstable. A wavefunction that exactly represents this system would represent the unbound state, whereas our approximate trial wavefunction represents bound states by definition.

Previously we discussed the importance of the interaction between the repulsive particles in calculating the energy of any chemical system. The critical mass is a prime example of this, for without this interaction all systems are stable and there is no  $m_{\text{cr}}$ . The results for the critical mass explain why all “normal” unit-charged chemical three-body systems (i.e. comprised of protons and electrons) are stable and why some of the more “exotic”, with different elementary particles are not necessarily stable (i.e. with muons and tauons).

For a more comprehensive discussion of the stability of small systems one should read the review article on the subject by Armour *et al.*<sup>[28]</sup> The review paper discusses advances towards calculating the stability of both specific systems such as  $\text{H}_2^+$ ,  $\text{H}^-$  and  $\text{Ps}^-$  and a general stability zone. What will later be referred to as Method 1 (Section 5.2.5) is also described in this paper, as a method for determining the complete stability zone for unit charge three-body systems.

The work in this chapter either enhances the previous results or provides an interpretation not present in these other works. Here the stability domain is improved by using more systems and calculating them as accurately as can be achieved with our Laguerre polynomial based wavefunction, which is more accurate than any previous study. This wavefunction was not used in these other works, and so we also



measure this wavefunction's performance for critical mass system. As can be seen from Chapter 4, our Laguerre polynomial wavefunction is efficient, even at large matrix sizes, and accurate for the more common symmetric systems.

This chapter aims to calculate a stability region for all unit-charge systems without using the BO approximation to treat the complete range between molecular and atomic. Two techniques are used to calculate the stability zone and this reveals the importance of certain symmetric and antisymmetric terms on the critical mass. This transition between the well-established norms of molecular and atomic behaviour manifests itself is also investigated with respects to the least fractional binding energy.

## 5.2. Method

A stable system is defined in this work as one in which there is at least one bound state below the most stable continuum threshold. This means that it must have a ground state energy lower (more stable) than the threshold energy. Three-body systems are ideal for studying stability because their most stable continuum threshold is the most stable two-body energy of its constituent parts (this is always the heavier oppositely charged pair). Two-body energies are known analytically as<sup>[17]</sup>:

$$E_{ij}^{2 \text{ body}} = -\frac{Z_i^2 Z_j^2}{2n} \frac{1}{m_i^{-1} + m_j^{-1}} \quad (5.2.1)$$

For the ground state case that we are interested in  $n = 1$ . Higher integer  $n$  are excited states. This threshold energy for a three-body system will be referred to as  $E_{th}$ .

To understand why some combination of particles, for example  $p^+ \mu^- e^-$ , are unstable, when all two-body combinations are stable, consider a two-body-like case with no interaction between its like-charged particles. This will be referred to as the non-interacting particle model. In this model the three-body energy is a trivial matter to calculate as it is the sum of two sets of two-body energies:

$$E^{\text{non-interacting}} = -\frac{Z_1^2 Z_3^2}{2n} \frac{1}{m_1^{-1} + m_3^{-1}} - \frac{Z_2^2 Z_3^2}{2n} \frac{1}{m_2^{-1} + m_3^{-1}} \quad (5.2.2)$$

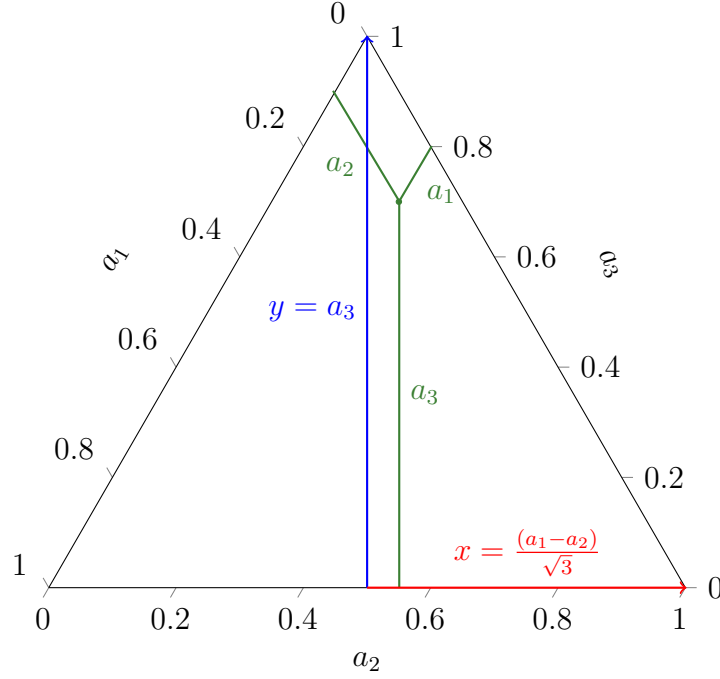
With this model, all systems are stable as any system will have an energy more stable than its  $E_{th}$ . The concept of a critical mass must therefore be a product of the repulsive interaction between the like-charged particles.

### 5.2.1. Ternary Diagram Explanation

This chapter uses ternary diagrams and it is therefore important that we clearly define them. A ternary diagram is a useful diagram on which to plot 3 quantities on a single 2-dimensional (2D) plot if, and only, if (iff) these quantities are related in such a way that the total of the three quantities sums to a single constant value. This means that there are, in truth, only 2 independent quantities and that the third can always be derived from the other two. Such diagrams are common in certain field of chemistry such as in phase diagrams.<sup>[130–135]</sup> The energy of three-body systems, with respect to either their mass or their charge, can be plotted on a ternary diagram with a few considerations. To plot all masses, only one set of charges is used i.e. unit charge of  $\pm, \pm, \mp$ . However to plot all charges, one set of masses is used e.g.  $e, e, \infty$ . This works because energy scales with both mass and charge (see Appendix B). Therefore the energy of a system with masses  $\{m_1^\pm, m_2^\pm, m_3^\mp\} = \{1^\pm, 1^\pm, 2^\mp\}$  can be used to find the energy of the system  $\{10^\pm, 10^\pm, 20^\mp\}$  by simply multiplying the energy by the scale factor 10. By choosing the 3 ternary diagram coordinates as normalised reciprocal masses that normalise to 1, we fulfil the condition that the ternary diagram coordinates sum to a constant value (in this case 1). This allows us to plot every mass, combination on a ternary diagram for a given charge. The same is true for every charge combination, with a given set of masses. These three ternary coordinates are as follows and are simply the three reciprocal masses normalised such that their sum is 1:

$$a_i = \frac{m_i^{-1}}{m_1^{-1} + m_2^{-1} + m_3^{-1}}, \quad i = 1, 2, 3 \quad (5.2.3)$$

The  $x$  and  $y$  axes of the ternary diagrams are defined as follows. The  $y$  axis is the  $a_3$  axis and it runs down the centre of the  $a_3$  axis as shown in Figure 5.2.1. The  $x$  axis runs perpendicular to this with positive values on right side of the triangle and negative on the left. In terms of normalised reciprocal masses ( $a_i$ ) the  $x$  axis has the



**Figure 5.2.1.:** Depicts the ternary diagrams coordinate system

coordinates of:  $x = \frac{a_1 - a_2}{\sqrt{3}}$ .

In this chapter the critical mass of systems with arbitrary, real, positive mass and fixed unit charge are investigated. This means that the charges must be  $Z_1 = -1$ ,  $Z_2 = -1$ ,  $Z_3 = +1$  or  $Z_1 = +1$ ,  $Z_2 = +1$ ,  $Z_3 = -1$  according to the inverse charge invariance of the Hamiltonian (see Appendix B). These two sets of charges for a given set of masses have the same energy and are identical wavefunctions. Although any charge could have been chosen the unit charge case has more literature available for comparison and this choice allows the use of a unique method for calculating critical mass, detailed later as Method 1. Two completely different methods to predict the critical masses of systems are used. These methods are derived from the literature<sup>[97,119,120,125,126,128,136,137]</sup> but have not been applied to any three-body wavefunctions composed of Laguerre polynomials prior to this work. The performance of the Laguerre polynomials to calculate critical masses will be informative from the aspect of Laguerre polynomial wavefunction performance. Prior to this chapter only bound systems in which a wavefunction is clearly defined have been studied. The critical mass lies on the boundary between two configurations of bound and unbound, making it difficult to accurately describe. The performance of the Laguerre polynomial wavefunction with the critical mass will judge its ability to

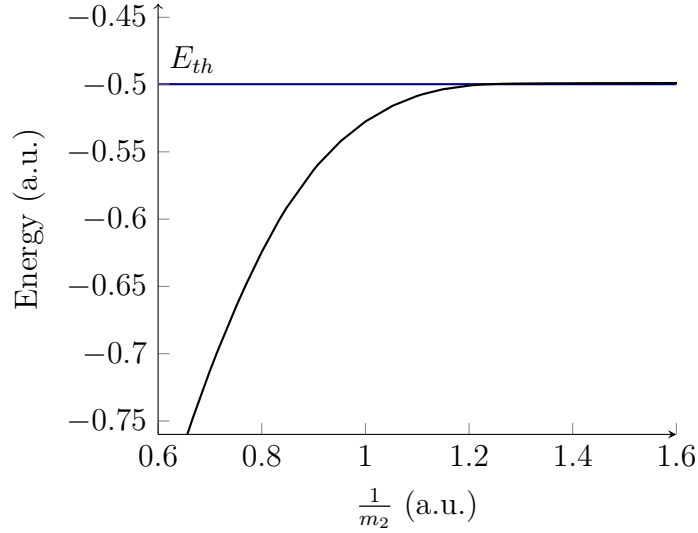
describe unusual configurations in extreme cases.

Hill *et al.*<sup>[138,139]</sup> proved that all symmetric systems are stable and therefore all systems of critical mass are asymmetric, as in  $m_1 \neq m_2$ . Therefore we use the full asymmetric wavefunction with all 3 non-linear variational parameters, as described in Section 4.2.8 on page 105.

### 5.2.2. Method 0: Energy as a Function of Mass

The most conceptually simple method would be to take a known, stable, three-body system such as the  $H^-$  atom (also known as (a.k.a)  $e^-e^-p^+$ , and reduce the mass of an electron until the energy of the bound three-body system is lower than the lowest continuum threshold. The following Figure 5.2.2. shows how for  $H^-$  decreasing the mass of  $m_2$  would effect the energy until it reaches the threshold energy (the horizontal line).

In Figure 5.2.2 it can be seen that as the mass of  $m_2$  decreases the energy of the system decrease until the mass reaches a critical point. At this point the energy then plateaus. This occurs at threshold energy and is the critical point we have been discussing as at this point the particle with mass  $m_2$  is no longer bound. This method simply varies  $m_2$  until the energy is equal to the threshold energy, to the desired number of digits. From Figure 5.2.2 it can also be seen that when the mass of  $m_2$  goes below a value ( $m_{cr}$ ) that particle no longer contributes to the total energy. From this critical mass and lower only the  $m_1$  and  $m_3$  particles contribute to the total energy. This method is convoluted and requires multiple calculations for a single  $m_{cr}$ . It is therefore not used in this chapter and the following two more efficient methods are used.



**Figure 5.2.2.:** The effect of changing the mass of  $m_2$  on the energy of “H<sup>-</sup>” using a 1078-term AC wavefunction.

### 5.2.3. Excess Binding Energy

We will deal with the concept of stability at various stages throughout this thesis. Indeed it is a central theme of 2 chapters. We therefore define a concept that is convenient to describe the energy of a system ( $E_0$ ) relative to its threshold energy ( $E_{th}$ ). This is the fraction of extra binding over its threshold energy it is given the symbol  $g$ . The threshold energy is the most stable continuum threshold. This is the two-body energy of the heavier oppositely charged pair of particles.

$$g = \frac{E_0 - E_{th}}{E_{th}} \quad (5.2.4)$$

Equation (5.2.4) defines the fraction of extra energy gained by adding a third particle.  $E_0$  is the three-body energy and  $E_{th}$  the threshold energy. With this  $g$  value and the  $E_{th}$  value the energy can be calculated by a simple calculation of  $E_0 = (1 + g) E_{th}$ . Additionally  $g$  is indicative of the stability. For example a  $g$  of zero means that there is no extra energy in the three-body system than that of the continuum threshold and the system is lying on the boundary between stable and unstable to dissociation of the third particle.

#### 5.2.4. Coordinate System

For future reference the  $r_1$ ,  $r_2$  and  $r_3$  are triangular coordinates that represent the sides of the triangle as shown previously Figure 4.2.1 on page 90.  $Z_i$  is the charge of the particle with the mass of  $m_i$ . For all three-body systems in this thesis  $m_3$  will be the uniquely charged particle and  $m_1$  and  $m_2$  will have a charge with the same sign i.e.  $m_1^\pm m_2^\pm m_3^\mp$ . Symmetric systems are defined as those that use the symmetric wavefunction where particle 1 is identical to particle 2 in terms of both mass and charge. Asymmetric is where particle 1 is not identical to particle 2.

#### 5.2.5. Method 1: Mass Symmetry Breaking Using the Symmetric Ground State Wavefunction

The first method, henceforth known as Method 1, is the method employed by Martin *et al.* [97] with some changes to aspects such as the form of the fitting function and the amount of data used to form it. One of the primary reasons many data points are possible in the calculations is the speed of modern computation and the efficient choice of methodology made possible by the Laguerre polynomial series solution method. The method relies on the variational technique but does not calculate critical mass directly. The method has several steps and it is based on the concept of using the wavefunction of a symmetric wavefunction (in which  $m_1^{-1}$  and  $m_2^{-1}$  are the average inverse masses of an arbitrary asymmetric wavefunction) as an approximate upper bound to the true asymmetric wavefunction. This is possible because of the convex nature of any property that enters the Hamiltonian linearly. [97,129,137] The following discussion demonstrates how Method 1 works, why ternary diagrams are so useful, and what makes it different from an exact method. In the first step of the Method 1 we rearrange the Hamiltonian into a symmetric part and an asymmetric part.

$$\hat{H} = -\frac{\nabla_1^2}{2m_1} - \frac{\nabla_2^2}{2m_2} - \frac{\nabla_3^2}{2m_3} + \hat{V} \quad (5.2.5)$$

$$\hat{H} = -\left(\frac{1}{2m_1} + \frac{1}{2m_2}\right) \frac{(\nabla_1^2 + \nabla_2^2)}{2} - \left(\frac{1}{2m_1} - \frac{1}{2m_2}\right) \frac{(\nabla_1^2 - \nabla_2^2)}{2} - \frac{\nabla_3^2}{2m_3} + \hat{V} \quad (5.2.6)$$

### 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

Where  $\hat{V}$  is the same three-body potential energy seen previously. We can see that this Hamiltonian can be broken into symmetric  $\hat{H}_{\text{sym}}$  and asymmetric  $\hat{H}_{\text{asym}}$  parts.

$$\hat{H} = \hat{H}_{\text{sym}} + \hat{H}_{\text{asym}} \quad (5.2.7)$$

$$\hat{H}_{\text{sym}} = - \left( \frac{1}{2m_1} + \frac{1}{2m_2} \right) \frac{(\nabla_1^2 + \nabla_2^2)}{2} - \frac{\nabla_3^2}{2m_3} + \hat{V} \quad (5.2.8)$$

$$\hat{H}_{\text{asym}} = - \left( \frac{1}{2m_1} - \frac{1}{2m_2} \right) \frac{(\nabla_1^2 - \nabla_2^2)}{2} \quad (5.2.9)$$

When  $m_1 = m_2$ , as in the system is symmetric, then the  $\hat{H}_{\text{asym}}$  becomes 0. Operating only with the symmetric Hamiltonian  $\hat{H}_{\text{sym}}$  on an arbitrary asymmetric system (i.e.  $\langle \psi_{\text{asym}} | \hat{H}_{\text{sym}} | \psi_{\text{asym}} \rangle$ ) would give the energy of a symmetric system where  $\frac{1}{m_1}$  and  $\frac{1}{m_2}$  are averaged. As the term  $\hat{H}_{\text{asym}}$  serves only to lower the energy it must therefore be that the energy of an arbitrary asymmetric system is lower than the energy of a symmetric system in which the inverse of these asymmetric masses are averaged. Therefore:

$$E(m_1, m_2, m_3) \leq E \left( \frac{1}{\frac{1}{2m_1} + \frac{1}{2m_2}}, \frac{1}{\frac{1}{2m_1} + \frac{1}{2m_2}}, m_3 \right) \quad (5.2.10)$$

This is where the use of  $a_i$  and the ternary diagram become very useful. An average of the  $a_i$  is an average of the reciprocal masses. If the normalised reciprocal mass of  $m_3$  is kept constant and we vary only  $a_1$  and  $a_2$  then due to the normalisation constraint  $a_1 + a_2 + a_3 = 1$  the average of  $a_1$  and  $a_2$  must not change. If we plot all  $a_i$  on the ternary diagram in which the average mass does not change, then this line is parallel with the  $x$  axis. This convenient concordance of average  $a_1, a_2$  with the  $x$  axis of our ternary diagram, when  $a_3$  is constant, makes plotting on the ternary diagram useful for this method. Therefore if we take a symmetric system as a lower bound to a critical mass we can calculate the critical mass by shifting along the  $x$  axis by some unknown amount. The key is to find out how much we must move along the  $x$  axis. We will call this amount critical delta ( $\delta_{\text{cr}}$ ).

To calculate  $\delta_{\text{cr}}$  the energy of the symmetric system must be changed by varying the masses until the energy is equal to the lowest continuum threshold. This following discussions details the mathematics behind the methods for identifying critical masses

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

in the method of Martin *et al.* [28,97]

$$E_0 = E_{th} \quad (5.2.11)$$

To facilitate this we first apply the following expression for the symmetric energy.

$$E_0^{\text{symmetric}} = E_{th}^{\text{symmetric}} g \quad (5.2.12)$$

The purpose of this is to allow us to express the symmetric average energy with a two-body energy, for which an exact expression is known in terms of its masses.  $g$  is the excess binding energy, as in the fraction of energy greater than the threshold energy as defined in Section 5.2.3. Another useful expression is:

$$\frac{2E_{th}^0}{E_{th}^0 + E_{th}^1} = \frac{E_{th}^{\text{average}}}{E_{th}^0} \quad (5.2.13)$$

Here we define  $E_{th}^0$  as the first continuum threshold (two-body energy of particles 3 and 2),  $E_{th}^1$  as the second continuum threshold (two-body energy of particles 3 and 1) and  $E_{th}^{\text{average}}$  as the first continuum threshold of the system with the averaged reciprocal masses. The expression in Eq. (5.2.13) is useful because it behaves in the way we desire i.e. is  $< 1$  for systems with  $E_{th}^1 > E_{th}^0$  or 1 when equal. Eq. (5.2.13) is known analytically and can be expressed in normalised reciprocal masses ( $a_i$ ). By using this Eq. (5.2.13) an expression for  $\delta_{\text{cr}}$  is calculated.

$$E_0 = E_{th} \leq E_0^{\text{average}} = E_{th}^{\text{average}} (1 + g) \quad (5.2.14)$$

$$E_0^{\text{average}} = E_{th} (1 + g) \frac{E_{th}^{\text{average}}}{E_{th}} \quad (5.2.15)$$

Equations (5.2.14) and (5.2.15) represents a trivial change but it incorporates a  $\frac{E_{th}^{\text{average}}}{E_{th}}$  term which we know exactly in normalised reciprocal mass as the form:

$$\frac{E_{th}^{\text{average}}}{E_{th}} = \frac{1 + a_3 - (a_2 - a_1)}{1 + a_3} \quad (5.2.16)$$

Therefore:

$$E_0 \leq E_{th} (1 + g) \frac{1 + a_3 - (a_2 - a_1)}{1 + a_3} \quad (5.2.17)$$



## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

It can be seen from this form that when:

$$(1 + g) \frac{1 + a_3 - (a_2 - a_1)}{1 + a_3} = 1 \quad (5.2.18)$$

then the energy is  $\leq E_{th}$  and that we have the critical mass condition. To calculate the critical delta we rearrange Eq. (5.2.18) such that  $(a_2 - a_1)$  becomes the subject, then substitute this into the general form of  $\delta$  in Eq. (5.2.19).

$$\delta = \frac{2|a_2 - a_1|}{\sqrt{3}} \quad (5.2.19)$$

$$\delta_{cr} = \frac{2g(1 + a_3)}{\sqrt{3}(1 + g)} \quad (5.2.20)$$

The symmetric energy can be used from any system to calculate a  $\delta_{cr}$ . Then by using this  $\delta_{cr}$  we calculate the normalised reciprocal mass at this point to get critical mass using a symmetric system's energy.

### 5.2.6. Method 2: Variational Principle for the Critical Mass

The next method uses the true asymmetric wavefunction to calculate the critical mass. The method is considerably simpler and relies upon accurate wavefunctions for the case where one particle is only just bound. We will refer to this as Method 2. This method is as complete as can be made within the non-relativistic time independent approximation, and with an exact wavefunction it will give an exact critical mass. It is also variational, and like Method 1, provides an upper bound to the critical mass. This method takes the Schrödinger equation and rearranges it as described in the following discussion:

$$\hat{H}\psi = E\psi \quad (5.2.21)$$

The following equations show a simplified view of the rearrangement. However this view is accurate in that only the complicated factors such as those associated with the transformation to perimetric coordinates are left out.

- At the critical mass when the energy of the three-body system is identical to that of the two-body system corresponding to the threshold energy (Equa-

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

tion (5.2.1))

$$E = E_{th} = \frac{1}{2} Z_1^2 Z_3^2 \frac{m_1 m_3}{m_1 + m_3} \quad (5.2.22)$$

- It then follows that the energy in the Schrödinger equation is not an unknown.

$$(\hat{T} + \hat{V}) \Psi = E_{th} \Psi \quad (5.2.23)$$

- The mass  $m_2$  is defined as the unknown with the property that it ensures  $E = E_{th}$ . This  $m_2$  is the value we desire to calculate. This  $m_2$  must be lighter than  $m_1$  for the threshold condition therefore  $m_1 \geq m_2$ .  $\frac{1}{m_2}$  is made the eigenvalue of the equation in place of the energy.

$$\hat{T}_i = - \frac{\nabla_i^2}{2m_i} \quad (5.2.24)$$

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 \quad (5.2.25)$$

$$\left( \hat{T}_1 - \frac{\nabla_2^2}{2m_2} + \hat{T}_3 + \hat{V} \right) \Psi = E_{th} \Psi \quad (5.2.26)$$

- It is a simple matter to make  $\frac{1}{m_2}$  the eigenvalue.

$$\left( \hat{T}_1 - \frac{\nabla_2^2}{2m_2} + \hat{T}_3 + \hat{V} \right) \Psi = E_{th} \Psi \quad (5.2.27)$$

$$\left( \hat{T}_1 + \hat{T}_3 + \hat{V} - E_{th} \right) \Psi = \frac{\nabla_2^2}{2m_2} \Psi \quad (5.2.28)$$

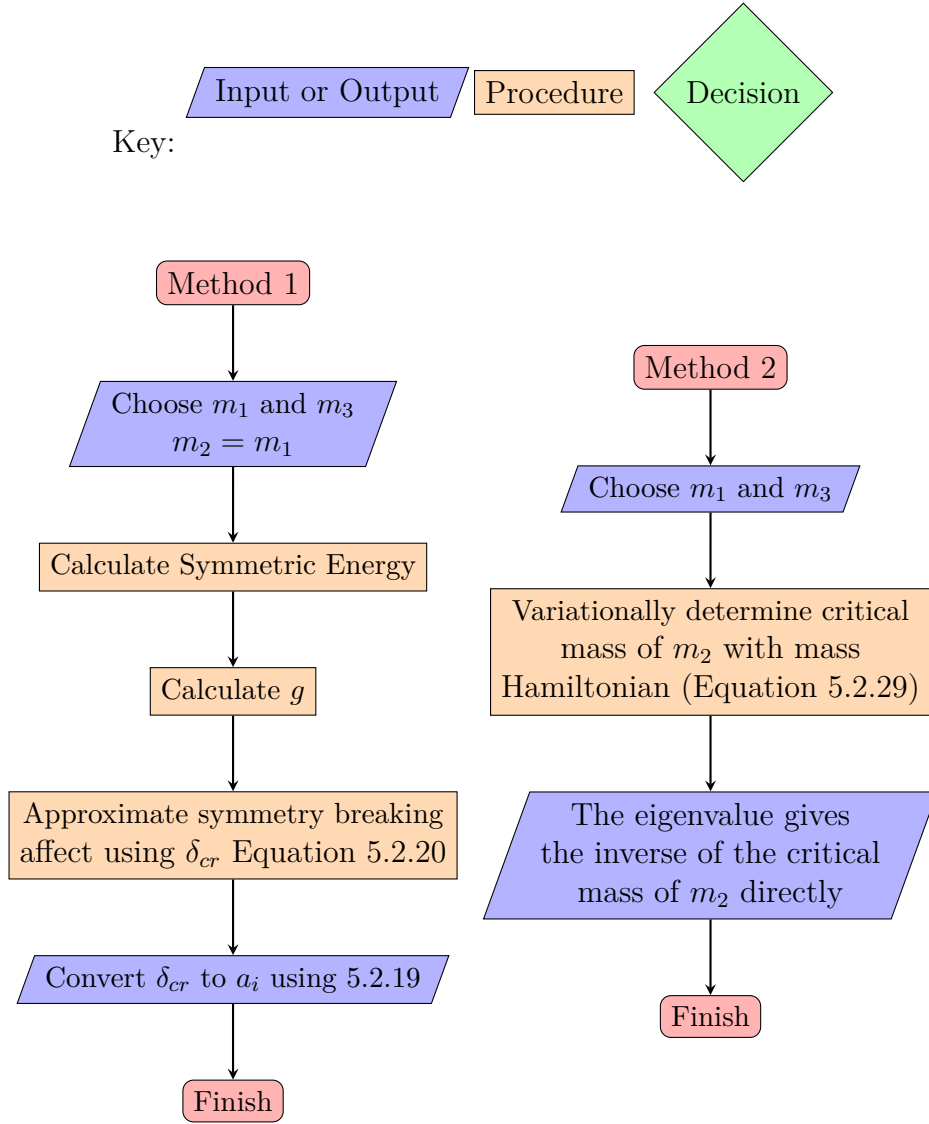
$$\left( \underbrace{\hat{T}_1 + \hat{T}_3 + \hat{V} - E_{th}}_{\text{Operator}} \right) \Psi = \underbrace{\frac{1}{m_{cr}}}_{\text{Eigenvalue}} \underbrace{\frac{\nabla_2^2}{2}}_{\text{Overlap}} \Psi \quad (5.2.29)$$

A simple rearrangement does not guarantee that an appropriate eigenproblem is formed, however Rebane and Kuzminskii showed that this is a valid, self-consistent, eigenproblem.<sup>[120]</sup> This method is superior in that it uses both the asymmetric wavefunction and the Hamiltonian, and that for an exact wavefunction it calculates the exact critical mass.

### 5.3. Implementation

The above methodologies are implemented in both Maple and C++. Maple was used to calculate the critical masses from energies as described in the Method 1 using the formalism described in Section 5.2. C++ was used to calculate these energies and to calculate the critical mass directly as described in Method 2 (Section 5.2.6). The C++ method relies on a series solution that is generated in Maple as detailed in the Programming Chapter 3. The same Laguerre polynomial-based  $AC$  or  $ABC$  wavefunction, as described in Section 4.2.8, is used throughout and in this chapter both the symmetric wavefunction and asymmetric wavefunctions are used. These use different numbering schemes as described in Appendix A. The parameters in the wavefunction are optimised using either the NROpt method or the BOBYQA method to locate minima. In both methods we use the general eigen-solving techniques as described in Section 4.2 on page 89. This involves the use of the Cholesky Decomposition and the MR3 diagonalisation method.

Figure 5.3.1 shows the general work flow for both methods. Method 1 is a more complicated procedure than Method 2. In this chapter both methods are investigated for the purpose of comparison. For the remainder of this chapter it will always be assumed that  $m_2 \leq m_1$ . This is enforced by switching the labels of the particles in the opposite case. This means that the lowest continuum threshold will now be the two-body energy between particles  $m_1$  and  $m_3$ , removing any ambiguity over which two-body energy should be used for the threshold energy.



**Figure 5.3.1.:** These flow charts detail the steps required for the computation of the critical mass.

## 5.4. Results & Discussion

### 5.4.1. Method 1: The Calculation of the Stability Domain

The wavefunction is used to calculate accurate energies,  $E_0$ , for a wide range of unit-charge, symmetric, three-body systems of the form  $\{m_1^\pm m_2^\pm m_3^\mp\}$ , involving the particles listed in Section 4.2.2 on page 90. There are 36 distinct, symmetric, singly charged ions ranging from the atomic to the molecular (72 including the anti-particle counterparts which, by charge inversion invariance (See Appendix B), have the same energy as those presented, e.g.,  $E(p^+p^+e^-) = E(p^-p^-e^+)$ ). Additionally, some hypothetical mass systems are calculated to ensure an even distribution of data points along the symmetric axis of the reciprocal mass fraction ternary diagram. Accurate values of the fractional additional binding of  $g$  (see Section 5.2.3) are calculated using these  $E_0$  values. The expression of Martin *et al.*<sup>[97]</sup> for the width of the stability band (Equation (5.2.20)) is used to determine an accurate lower bound to the stability zone of three-body coulomb systems, and the results are presented on a ternary diagram.<sup>[97,137]</sup>

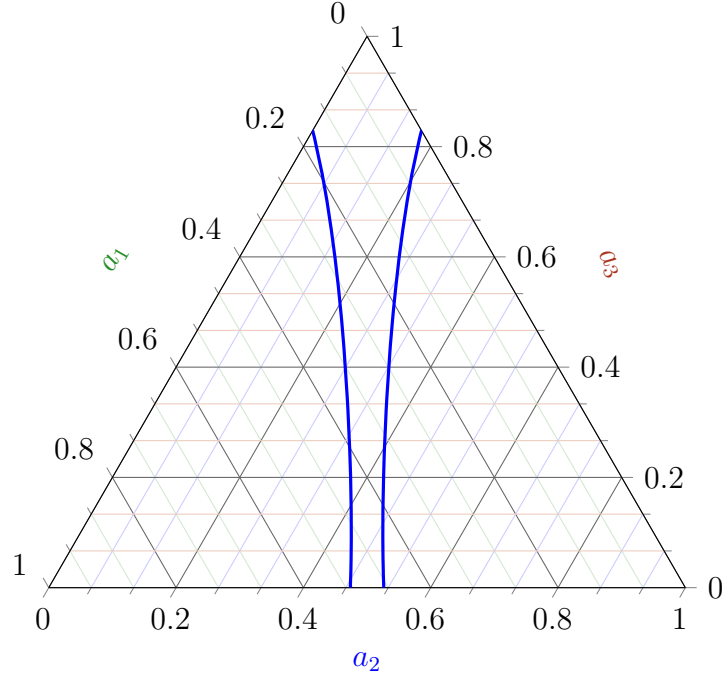
The ground state energy of a wide-range of symmetric systems of the form  $\{m_1^\pm m_2^\pm m_3^\mp\}$  were calculated in a series ranging from  $\{(m_1 = m_2) \gg (m_3 = 1)\}$  to  $\{m_3 \gg (m_1 = m_2 = 1)\}$ . For consistency, the charges on each particle were set as  $\{m_1^- m_2^- m_3^+\}$ , although reversing the charges would produce the same  $E_0$  due to charge inversion invariance. The calculated energies for all the systems considered are given in Table 5.4.1. To determine the stability of three-body systems, the lowest energy threshold is considered. Using the  $g$  values directly, the lower bound of  $\delta_{cr}$  can be calculated using Eq. (5.2.20) for each  $a_3$  value forming the central axis and the results presented on a mass fraction ternary diagram (Figure 5.4.1).

**The general shape of a stability zone** There exists a zone of stability for three-body systems.<sup>[28,97,120]</sup> On the inverse normalised mass fraction ternary diagram it is possible to deduce the basic shape of such a zone before any calculations. This is useful as it can be used as a guide for the successful plotting of the diagram, ensuring that we capture all regions of stability. Referring back to Section 5.2.1, any set of three masses can be plotted onto the ternary diagram within the unit charge regime.

5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

System	Energy ( $E_h$ )	$\frac{m_1}{m_3}$	$a_3$	$g$
$\infty^+ \infty^+ e^-$	-0.602 148 905	$\infty$	1	0.204 286 537
$t^+ t^+ e^-$	-0.599 506 903	$5.4969 \times 10^3$	0.999 636 292	0.199 231 931
$d^+ d^+ e^-$	-0.598 788 784	$3.6705 \times 10^3$	0.999 455 409	0.197 903 841
$\tau^+ \tau^+ e^-$	-0.598 680 507	$3.4772 \times 10^3$	0.999 425 147	0.197 705 365
$p^+ p^+ e^-$	-0.597 139 063	$1.8362 \times 10^3$	0.998 911 951	0.194 928 550
$\mu^+ \mu^+ e^-$	-0.585 126 098	$2.0677 \times 10^2$	0.990 420 001	0.175 911 924
$t^+ t^+ \mu^-$	-112.972 859	$2.6585 \times 10^1$	0.930 033 077	0.133 852 490
$d^+ d^+ \mu^-$	-109.816 937	$1.7752 \times 10^1$	0.898 742 754	0.122 060 128
$\tau^+ \tau^+ \mu^-$	-109.331 147	$1.6817 \times 10^1$	0.893 711 158	0.120 408 873
$11.3^+ 11.3^+ 1^-$	-0.508 912 278	$1.1300 \times 10^1$	0.849 624 060	0.107 897 524
$p^+ p^+ \mu^-$	-102.223 513	8.8802	0.816 180 581	0.100 118 955
$6^+ 6^+ 1^-$	-0.466 087 497	6.0000	0.750 000 000	0.087 537 493 6
$4.7^+ 4.7^+ 1^-$	-0.445 261 313	4.7000	0.701 492 537	0.079 995 526 3
$3.7^+ 3.7^+ 1^-$	-0.422 360 557	3.7000	0.649 122 807	0.073 024 118 3
$t^+ t^+ p^-$	-734.528 908	2.9937	0.599 496 730	0.067 324 900 0
$2.4^+ 2.4^+ 1^-$	-0.374 811 142	2.4000	0.545 454 545	0.061 964 903 1
$d^+ d^+ p^-$	-647.474 584	1.9990	0.499 875 907	0.058 052 104 3
$\tau^+ \tau^+ p^-$	-635.051 596	1.8937	0.486 351 733	0.056 991 143 3
$t^+ t^+ \tau^-$	-1122.214 36	1.5809	0.441 476 486	0.053 785 891 2
$t^+ t^+ d^-$	-1158.683 73	1.4976	0.428 179 601	0.052 927 692 6
$d^+ d^+ \tau^-$	-936.125 281	1.0556	0.345 464 277	0.048 526 893 0
$\tau^+ \tau^+ \tau^-$	-911.030 930	1.0000	0.333 333 333	0.048 020 280 8
$t^+ t^+ t^-$	-1440.221 31	1.0000	0.333 333 333	0.048 020 280 8
$e^+ e^+ e^-$	-0.262 005 070	1.0000	0.333 333 333	0.048 020 280 8
$\mu^+ \mu^+ \mu^-$	-54.174 338 8	1.0000	0.333 333 333	0.048 020 280 8
$p^+ p^+ p^-$	-481.081 310	1.0000	0.333 333 333	0.048 020 280 8
$d^+ d^+ d^-$	-961.685 147	1.0000	0.333 333 333	0.048 020 280 8
$\tau^+ \tau^+ d^-$	-935.260 852	$9.4733 \times 10^{-1}$	0.321 419 184	0.047 558 670 6
$d^+ d^+ t^-$	-1150.610 15	$6.6773 \times 10^{-1}$	0.250 300 147	0.045 591 012 4
$\tau^+ \tau^+ t^-$	-1113.316 62	$6.3256 \times 10^{-1}$	0.240 284 132	0.045 430 702 2
$p^+ p^+ \tau^-$	-627.929 517	$5.2806 \times 10^{-1}$	0.208 880 332	0.045 137 029 2
$p^+ p^+ d^-$	-639.557 763	$5.0025 \times 10^{-1}$	0.200 079 431	0.045 115 069 1
$p^+ p^+ t^-$	-719.649 173	$3.3403 \times 10^{-1}$	0.143 114 052	0.045 703 543 4
$1^+ 1^+ 5^-$	-0.436 497 015	$2.0000 \times 10^{-1}$	0.090 909 090 9	0.047 592 835 3
$\mu^+ \mu^+ p^-$	-97.566 992 7	$1.1261 \times 10^{-1}$	0.053 303 521 4	0.050 005 962 0
$\mu^+ \mu^+ \tau^-$	-102.672 896	$5.9465 \times 10^{-2}$	0.028 873 944 0	0.052 176 133 2
$\mu^+ \mu^+ d^-$	-102.991 921	$5.6333 \times 10^{-2}$	0.027 394 749 0	0.052 325 179 0
$\mu^+ \mu^+ t^-$	-104.944 126	$3.7615 \times 10^{-2}$	0.018 460 446 3	0.053 272 072 9
$e^+ e^+ \mu^-$	-0.525 054 806	$4.8363 \times 10^{-3}$	0.002 412 332 41	0.055 188 291 3
$e^+ e^+ p^-$	-0.527 445 881	$5.4462 \times 10^{-4}$	0.000 272 234 379	0.055 466 274 1
$e^+ e^+ \tau^-$	-0.527 589 838	$2.8759 \times 10^{-4}$	0.000 143 775 251	0.055 483 136 3
$e^+ e^+ d^-$	-0.527 598 325	$2.7244 \times 10^{-4}$	0.000 136 203 302	0.055 484 130 9
$e^+ e^+ t^-$	-0.527 649 048	$1.8192 \times 10^{-4}$	0.000 090 951 730 3	0.055 490 076 0
$e^+ e^+ \infty^-$	-0.527 751 017	0.	0.	0.055 502 033 1

**Table 5.4.1.:** Total energy (in electron atomic units), mass ratio  $\left(\frac{m_1}{m_3}\right)$ , reciprocal mass of the uniquely charged particle ( $a_3$ ), and relative excess binding energy ( $g$ ), for some symmetric configurations of the form  $\{m_1^\pm m_2^\pm m_3^\mp\}$ .

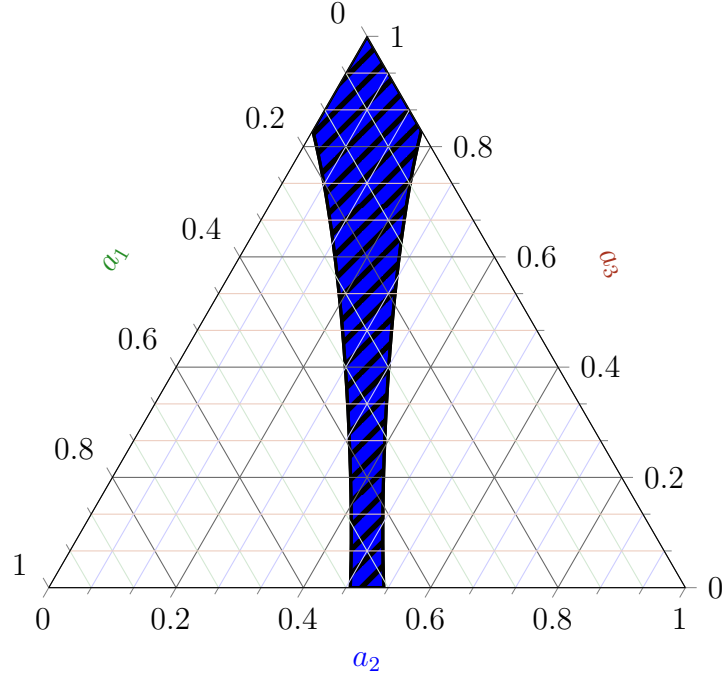


**Figure 5.4.1.:** The stability boundary as calculated by the data in Table 5.4.1

**The central axis (symmetric systems)** Certain points on the ternary diagram are stable and have been observed in nature, for example  $e^-e^-p^+$ ,  $p^+p^+e^-$ ,  $e^-e^-e^+$ ,  $d^+d^+e^-$ ,  $p^+p^+\mu^-$ . These points all lie along the central axis of the ternary diagram (Section 5.2.1). Hill *et al.* <sup>[138]</sup> proved that all symmetric systems are stable. Therefore the stability zone must contain the central axis of the ternary diagram.

**Single crossing from stable to unstable** Moving in a straight line from a point on the central axis of the ternary diagram to a lower corner causes the system to cross the boundary only once. Such a line will have a constant  $\frac{m_1}{m_3}$  ratio for a line towards the right corner and constant  $\frac{m_2}{m_3}$  for the other. Moving along the line towards the bottom right corner means that only  $m_2$  changes, to be precise the mass decreases as we move closer to this corner. The definition of  $E_{th}$  does not depend upon  $m_2$  and therefore  $E_{th}$  does not change. However as the mass of  $m_2$  decreases the energy of the three-body system increases. It follows then that such a line will only ever cross the boundary between  $E_0 \geq E_{th}$  once.

**The boundary is concave** When two arbitrary points on the boundary ( $M' = \{a'_1, a'_2, a'_3\}$  and  $M'' = \{a''_1, a''_2, a''_3\}$ ) are taken, the points between them are unstable. This is discussed in more depth by Martin *et al.* <sup>[97]</sup> The concept will be briefly



**Figure 5.4.2.:** The blue shaded region is stable with at least one bound state more stable than the lowest continuum threshold.

discussed here. Taking two scaled inverse masses such that the threshold energy remains the same ( $N' = \{x'_1, x'_x, a'_3\}$  and  $N'' = \{x''_1, x''_2, x''_3\}$ ), where  $x_i = \frac{a_i}{a_1 + a_3}$ . Any system on the line between these two points can be represent with  $N(\lambda)$  where  $\lambda$  is from 0 to 1 and  $N(\lambda) = \lambda N' + (1 - \lambda) N''$ . As  $\lambda$  shifts the inverse masses and the inverse masses enters linearly into the Hamiltonian (See linear attributes of the Hamiltonian Appendix C) then, from a general mathematical theorem<sup>[104]</sup> it follows that:

$$E_0(\lambda) \geq \lambda E_0(1) + (1 - \lambda) E_0(0) \quad (5.4.1)$$

Since  $N'$  and  $N''$  are on the boundary and the threshold energy does not change, and the  $M \rightarrow N$  transformation is conical, thus preserving straight lines in  $N$  space when converting to  $M$  space then any system on the line between two arbitrary boundary systems is unstable.

**The stability domain** From the above arguments, the entirety of the inside of the boundary in Figure 5.4.1 is stable. This is shown in Figure 5.4.2 as the blue shaded region. From the stability zone in Figure 5.4.2, it can be deduced that all systems inside the blue shaded region are stable with respect to dissociation and therefore contain at least one bound state. To characterise this stability zone a function was



## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

fitted to the  $g$  data along the  $a_3$  such that for a given  $a_3$  we predict a  $g$  value and thus a  $\delta_{cr}$ . This fitted function has the form:

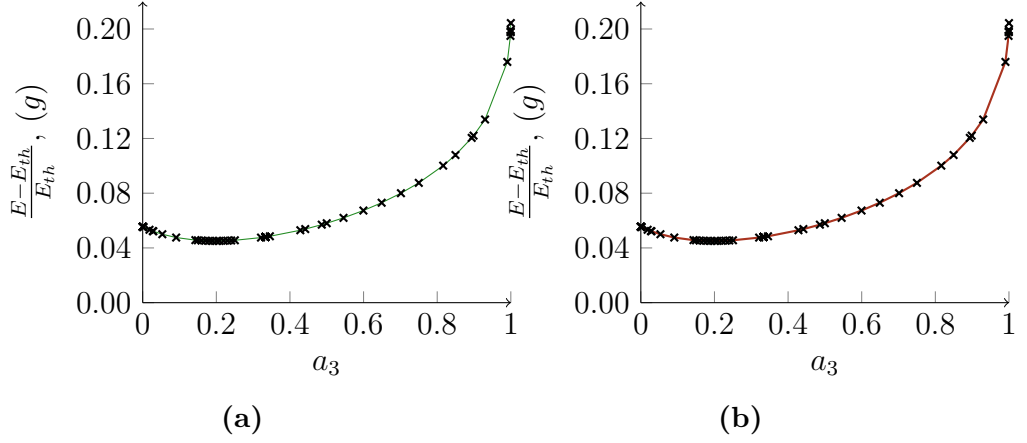
$$g(a_3) = \sum_{i=0}^{10} C_i (1 - a_3)^{\frac{i}{2}} \quad (5.4.2)$$

- $c_0 = 0.205\,340\,759$
- $c_1 = -0.326\,670\,401$
- $c_2 = -0.383\,300\,516$
- $c_3 = -2.048\,690\,69$
- $c_4 = 12.366\,540\,9$
- $c_5 = -46.866\,686\,4$
- $c_6 = 110.492\,236$
- $c_7 = -162.933\,844$
- $c_8 = 146.384\,325$
- $c_9 = -73.345\,924\,5$
- $c_{10} = 15.745\,569\,1$

Using our data to describe the stability zone more accurately the stability zone reaches the sides of the ternary diagram at  $a_3 = 0.8292(1)$ , corresponding to  $(m_1 = \infty, m_2 = 4.85(4), m_3 = 1)$ , and the lower edge of the stability zone at  $a_3 = 0$  corresponds to a mass ratio  $\frac{m_1}{m_2} = 1.11(1)$ , i.e.,  $(m_1 = 1.111, m_2 = 1, m_3 = \infty)$ . However, because  $\delta_{cr}$  is a lower bound, the stability region may be larger.

**Limitations of Method 1** The inaccuracies in the present result are due to underestimation of the width of the stability band. The lower limit of the stability band (Equation (5.2.20)) is derived by rewriting the Hamiltonian as a sum of symmetric terms (symmetric under  $1 \leftrightarrow 2$  exchange) and anti-symmetric terms, and then applying the variational principle to this Hamiltonian using the symmetric ground state wavefunction of the symmetric Hamiltonian as a trial wave function.<sup>[28,97]</sup> This

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems



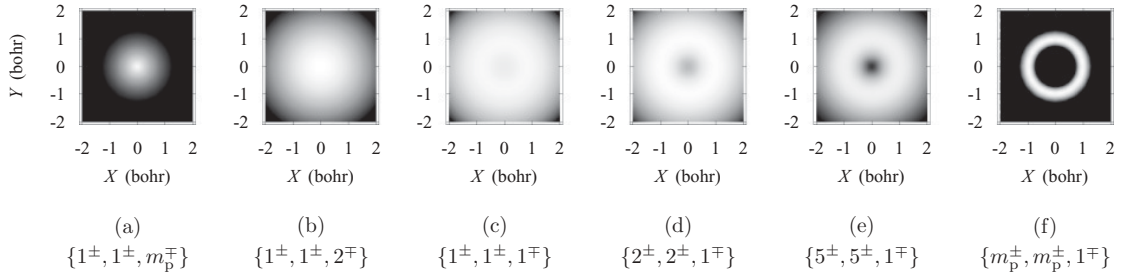
**Figure 5.4.3.:** (a) Plot of  $g$  against  $a_3$  for a range of symmetric systems, (b) the function  $g(a_3)$  (red line) compared with the raw data of  $g$  versus  $a_3$  (green crosses)

underestimate would indicate that the anti-symmetric (symmetry-breaking) terms are not small.<sup>[140]</sup> Clearly, a more accurate value could be obtained by solving the Schrödinger equation for three unequal mass particles directly.<sup>[72]</sup> Section 5.4.2 uses such a method to predict a more precise bound based on determining the value of the third particle mass that corresponds to the critical binding energy, i.e., when  $E_0 = E_{th}$ .

**The Atom to Molecule Transition** A plot of  $g$  against  $a_3$  for a range of symmetric systems yields the graph depicted in Figure 5.4.3(a). Particularly interesting is that  $g(a_3)$  does not decrease monotonically to  $a_3 = 0$  corresponding to  ${}^\infty\text{H}^-$  (as in  $\text{H}^-$  with infinite nuclear mass), but contains a minimum. Based on the data in Table 5.4.1, this minimum is estimated to occur at  $a_3 \approx 0.2$ , which corresponds to the system  $(\text{p}, \text{p}, \text{d})^\pm$ , with a mass ratio of  $\approx 0.5$ . To investigate this further, the data points listed in Table 5.4.1 and seven additional points having  $a_3$  values between the data points  $(\tau, \tau, \text{t})^\pm$  and  $(\text{p}, \text{p}, \text{t})^\pm$  in increments of 0.01, were fitted to a functional form, and the function differentiated to determine the minimum as accurately as possible.

**Minimum in  $g(a_3)$**  Differentiating the fitted function of  $g(a_3)$  to determine the minimum provides the symmetric system that has the least fractional binding energy gained by association with a third particle. However, the minimum is sensitive to the functional form and so additional data points were calculated about the minimum

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems



**Figure 5.4.4.:** Reprinted from “On the emergence of molecular structure”<sup>[100]</sup> by permission of author Mátyus. Original caption read: Transition from an atomic- to a molecular-type distribution of the particles in  $\{a^\pm, a^\pm, b^\mp\}$ -type coulombic systems. In each graph the normalized density plot of  $\rho_a(R)$  is shown for  $R = (X, Y, 0)$  in terms of  $X, Y \in [-2, 2]$  Bohr. The assembled particles are specified as  $\{a^\pm, a^\pm, b^\mp\}$ , where  $m_a$  and  $m_b$  are the masses and  $m_p = 1836.152\,672\,47$ <sup>[94]</sup> in units of the mass of an electron. The centre of mass is the centre of each plot.

to estimate the accuracy. Based on this analysis, and comparing with alternative functional forms, the minimum is estimated to occur at  $a_3 = 0.197(1)$ , corresponding to the system ( $m_1 = m_2 = 1$ ,  $m_3 = 2.04(1)$ ) which has a mass ratio of  $\frac{m_1}{m_3} = 0.491(3)$ .

**Significance of the minimum in  $g(a_3)$  function** A recent study by Mátyus *et al.*<sup>[100]</sup> into the emergence of molecular structure in coulombic three-particle systems considered the distribution of the identical particles in the ground state and concluded that a node in the particle density at the centre-of-mass first appears when  $0.4 < \frac{m_1}{m_3} < 0.8$ .<sup>[100]</sup> It is suggested that such a transition in density distribution could determine the crossover point between molecular and atomic behaviour. The results presented here, suggest that this cross-over point causes a destabilisation and thus corresponds to the system with the least fractional gain in binding energy by association with a third particle.

### 5.4.2. Method 2: The Variational Stability Domain

The variational method described in Section 5.2.6 (Method 2) to determine the critical mass of a system with a single calculation, was used to calculate the minimum mass required of a third particle in order to bind to selected two-body atomic systems. These are given in Table 5.4.2 in electron atomic units. The masses presented are considered accurate to at least 2 s.f., with some cases reaching 3 s.f., and were

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

calculated using, at most, the 8436-term wavefunction. The rate of convergence, even for similar systems, is not uniform. This work has been published in the Journal of Chemical Physics 2014.<sup>[136]</sup> In this paper, the masses were converged to 2 s.f. using a 3654-term wavefunction.

**Achieving Better Results** More converged digits than those reported in the Journal of Chemical Physics paper<sup>[136]</sup> are obtainable by using a wavefunction with more terms, as can be seen in Table 5.4.3 where we show the rate of convergence of the critical mass and the energy side by side as the number of terms increases. The accuracy of this type of method in which the mass is determined variationally is several orders of magnitude less converged than that achieved for the energy using the same methodology with much smaller matrices. This however is not an indication of more error in the wavefunction just that the critical masses are more sensitive to such error than the energies. Our results are well converged because we know the energy is converged and because we achieve results on par with other literature values.<sup>[120,128]</sup>

Therefore the data presented in Table 5.4.2 supersede those in the published work and are more accurate to a maximum of 3 s.f., and were achieved, for some systems, using a 8436-term wavefunction. For other systems however, due to limitations on computational time, only the lower wavefunction sizes in the paper are presented here. Critical mass for all possible combinations of particles in Section 4.2.2 on page 90 have been determined to ensure an accurate stability boundary.

**Comparison to Literature Values** Rebane and Kuzminskii<sup>[120]</sup> had similar successes to our own results in terms of convergence and reported critical mass values to 3 s.f. with a significantly smaller and more flexible exponential basis, but with many more (60) non-linear variational parameters. To ensure that our Laguerre polynomial-based wavefunction was capable of producing values at least on par with those of Rebane and Kuzminskii,<sup>[120]</sup> a larger wavefunction was used here rather than the wavefunction used in the J. Chem. Phys. work. However because of the quasi-orthogonality<sup>[141]</sup> of the wavefunction many non-diagonal elements of the Hamiltonian and overlap matrices are zero and it is quite efficient to use a many term Laguerre polynomial-based wavefunctions. Table 5.4.4 compares our results to those

5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

System	Critical Mass ( $m_e$ )	$\frac{m_1}{m_3}$
$e^- m_{cr}^- t^{+a}$	0.796	0.000 181 920 007
$e^- m_{cr}^- d^{+a}$	0.796	0.000 272 443 711
$e^- m_{cr}^- \tau^{+a}$	0.796	0.000 287 591 850
$e^- m_{cr}^- p^{+a}$	0.796	0.000 544 617 022
$e^- m_{cr}^- \mu^{+a}$	0.796	0.004 836 331 66
$\mu^- m_{cr}^- t^{+a}$	165	0.037 615 287 6
$\mu^- m_{cr}^- d^{+c}$	165	0.056 332 718 7
$\mu^- m_{cr}^- \tau^{+c}$	165	0.059 464 873 3
$\mu^- m_{cr}^- p^{+c}$	165	0.112 609 527
$p^- m_{cr}^- t^{+c}$	1440	0.334 032 906
$p^- m_{cr}^- d^{+b}$	1400	0.500 248 248
$p^- m_{cr}^- \tau^{+c}$	1400	0.528 062 543
$\tau^- m_{cr}^- t^{+c}$	2600	0.632 563 151
$d^- m_{cr}^- t^{+c}$	2740	0.667 734 285
$\tau^- m_{cr}^- d^{+c}$	2460	0.947 327 650
$e^- m_{cr}^- e^{+a}$	0.698	1.000 000 00
$d^- m_{cr}^- \tau^{+a}$	2530	1.055 600 99
$t^- m_{cr}^- d^{+a}$	3470	1.497 601 70
$t^- m_{cr}^- \tau^{+a}$	3410	1.580 869 83
$\tau^- m_{cr}^- p^{+a}$	2020	1.893 715 08
$d^- m_{cr}^- p^{+a}$	2090	1.999 007 50
$t^- m_{cr}^- p^{+a}$	2580	2.993 717 03
$p^- m_{cr}^- \mu^{+a}$	397	8.880 243 31
$\tau^- m_{cr}^- \mu^{+a}$	426	16.816 650 6
$d^- m_{cr}^- \mu^{+a}$	428	17.751 673 0
$t^- m_{cr}^- \mu^{+a}$	438	26.584 935 6
$\mu^- m_{cr}^- e^{+a}$	2.20	206.768 284
$p^- m_{cr}^- e^{+a}$	2.21	1836.152 67
$\tau^- m_{cr}^- e^{+a}$	2.21	3477.150 00
$d^- m_{cr}^- e^{+a}$	2.21	3670.482 97
$t^- m_{cr}^- e^{+c}$	2.22	5496.921 53

<sup>a</sup> 8436-term wavefunction

<sup>b</sup> 5456-term wavefunction

<sup>c</sup> 3654-term wavefunction

**Table 5.4.2.:** The critical masses as calculated with Method 2

### 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

Omega	Number of Terms	Energy ( $E_h$ )	$m_{cr}$ ( $m_e$ )
10	286	-0.527 750 986 748 39	2.268
15	816	-0.527 751 012 081 99	2.236
21	2024	-0.527 751 016 360 59	2.222
26	3654	-0.527 751 016 528 73	2.214
30	5456	-0.527 751 016 540 82	2.213
35	8436	-0.527 751 016 543 97	2.208
Literature:		-0.527 751 016 544 37 <sup>[120]</sup>	2.203

**Table 5.4.3.:** A comparison of convergence between the energy (of  $^\infty\text{H}^-$ ) and the critical mass. Omega refers to the Pekeris shell as discussed in Section 4.2.10.

System	$m_{cr}$ ( $m_e$ ) (this work)	$m_{cr}$ ( $m_e$ ) (literature) <sup>[120]</sup>
$p^- m_{cr}^- e^{+a}$	2.208	2.203
$p^- m_{cr}^- \mu^{+a}$	396.6	395.8
$e^- m_{cr}^- e^{+a}$	0.6983	0.6969
$\mu^- m_{cr}^- p^{+b}$	164.8	164.2
$e^- m_{cr}^- p^{+a}$	0.7961	0.7950

<sup>a</sup> 8436-term wavefunction

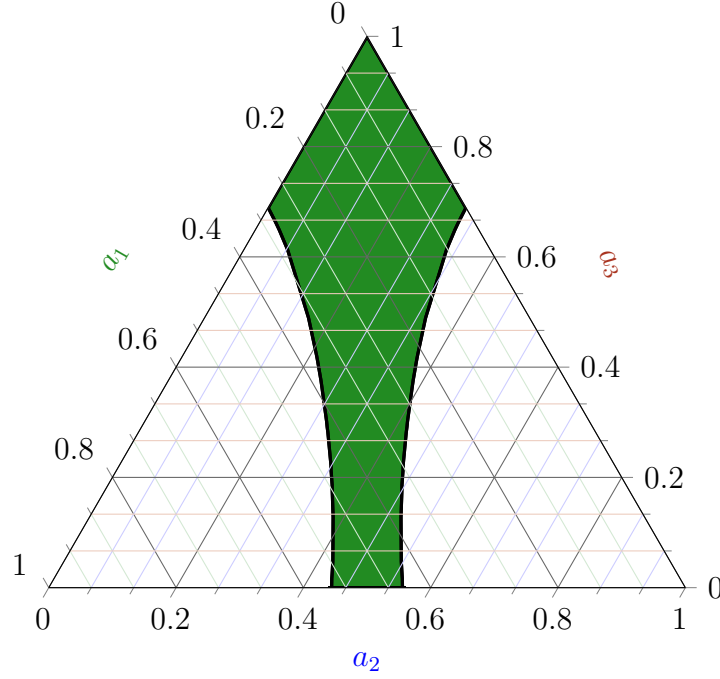
<sup>b</sup> 3654-term wavefunction

**Table 5.4.4.:** A comparison of our results against the literature of Rebane and Kuzminskii.<sup>[120]</sup>

of Rebane and Kuzminski.<sup>[120]</sup> We use these data to judge that our wavefunction is converged to between 2 s.f. and 3 s.f..

**The Stability Boundary of Method 2** A stability boundary was determined using the values in Table 5.4.2 and plotted on a reciprocal mass ternary diagram (Figure 5.4.5, green shaded region). This is referred to here as the “exact” stability boundary, as it is calculated using the exact threshold energy, with a precision determined by the accuracy of the threshold mass calculated variationally. It is to be compared with the lower bound stability calculated using the wavefunction of the symmetric systems (Figure 5.4.2 blue shaded region). The proofs provided by Martin and Armour,<sup>[28,97]</sup> as described in Section 5.4.1 regarding the topology of the stability domain remain valid. Therefore all systems enclosed within the green boundary (shaded green) are stable to dissociation.

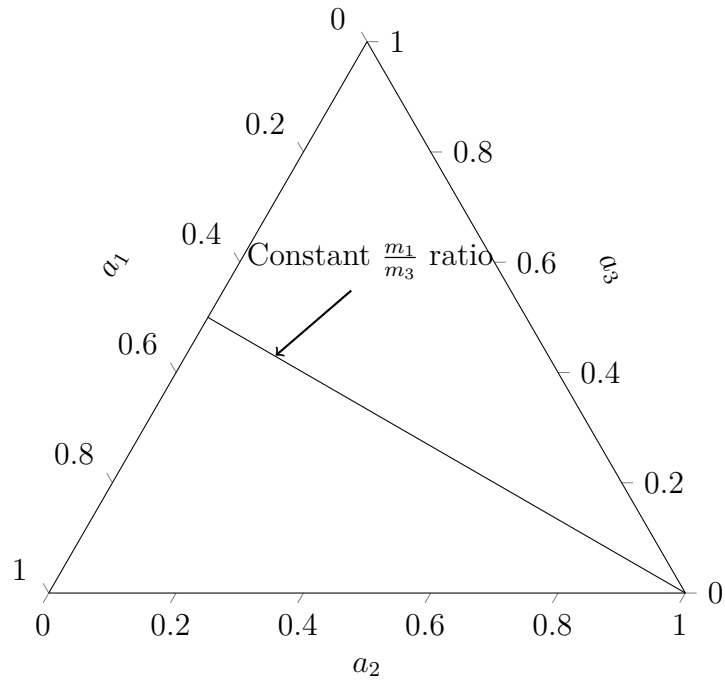
**Predicting the Stability of Any Three-body Unit Charge System a Priori** Predicting the stability of a particular three-body system simply requires the calculation



**Figure 5.4.5.:** The green shaded zone depicts the most accurate stability zone we have calculated for unit charge systems. Systems in the zone have at least one bound state.

of its position on the ternary diagram, using the definitions of the reciprocal mass given in Eq. (5.2.3). If a three-body system falls within the stability domain it is stable to dissociation. Given that the critical masses used to define the stability boundary are converged to 2 s.f. to 3 s.f., the error on the boundary is at best  $\pm 0.0005$  in scaled units. Within this accuracy, it would be useful to be able to determine the threshold mass of the third particle for stability with any particle pair. In order to do this, it is necessary to calculate the crossing point of the stability boundary with the line of constant mass ratio for the three-body system (Figure 5.4.6).

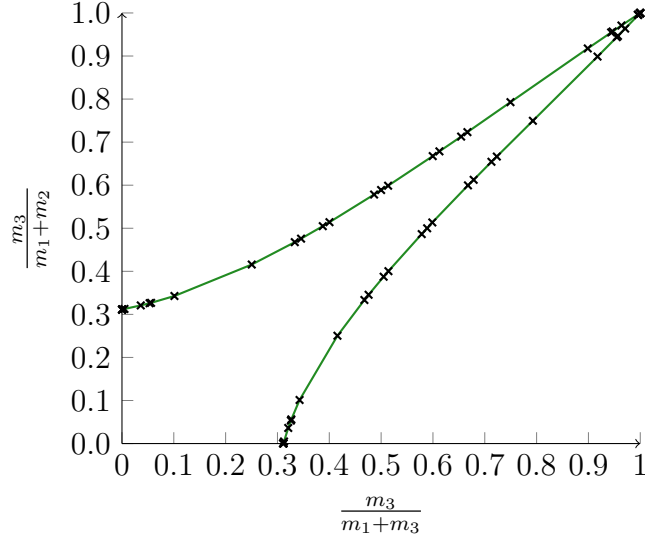
The coordinates in the ternary diagram, are  $x = \frac{(a_2 - a_1)}{\sqrt{3}}$  and  $y = a_3$ . Following the procedure in Section 5.4.1, it is possible to fit the boundary to a function of  $a_3$ , the reciprocal mass of the uniquely charged particle, and express the line of constant mass ratio  $\frac{a_1}{a_3}$  in these coordinates, and solve simultaneously. However computing the intercept between the line in Figure 5.4.6 and the boundary line in Figure 5.4.5 is difficult as the dependent value  $m_2$  is difficult to separate out of the  $a_i$  coordinates. Therefore, a much simpler method is to use the coordinate system,<sup>[120]</sup>  $x = \frac{m_3}{m_1 + m_3}$  and  $y = \frac{m_3}{m_2 + m_3}$  (as in barycentric coordinates for centre of mass of the two-body masses), as this eliminates  $m_2$  from the independent variable. Using this coordinate



**Figure 5.4.6.:** If  $m_1$  and  $m_3$  are kept fixed and  $m_2$  varied then tracing these systems on the ternary diagram gives a straight line from the lower right corner. This line on the ternary diagram represents the line with a given constant  $\frac{m_1}{m_3}$  mass ratio. Different values have different lines but they are all straight and all go through the lower right corner. The intercept of this line with the boundary gives the critical mass for a specific mass ratio.



## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems



**Figure 5.4.7.:** The stability boundary shown in barycentric coordinates, with the fitted function in green and calculated known systems as black crosses.

system, the stability boundary is fitted to a function of the form used in Section 5.4.1, i.e.

$$y = f(x) = \sum_{i=0}^5 c_i x^i \quad (5.4.3)$$

This results in an excellent fit with an  $R^2$  value of 0.999 999 83. The optimised values for this fit are:

- $c_0 = 0.310\,672\,03$
- $c_1 = 0.677\,996\,13 \times 10^{-2}$
- $c_2 = 0.195\,965\,44$
- $c_3 = -0.609\,288\,42 \times 10^{-1}$
- $c_4 = 1.359\,881\,3$
- $c_5 = -0.812\,377\,64$

The green line in Figure 5.4.7 is the fitted function and the crosses represent the calculated data points showing that the fitted function portrays the data accurately over the whole extent of the data. This boundary in Figure 5.4.7 shows much the same shape in these barycentric coordinates as it does with normalised reciprocal mass coordinates used with the ternary diagram in Figure 5.4.5. Substituting  $x = \frac{m_3}{m_1+m_3}$  into  $y = f(x)$  for a given mass pair  $m_1$  and  $m_3$  provides the lightest  $m_2$  mass possible

for binding to the two-body system. This is the first instance the author knows of where in which a simple formulae makes easy calculations of the prediction stability for any unit charge system.

### 5.4.3. Comparison of Lower Bound and Variational Methods of the Stability Domain

It is clear from Figure 5.4.5 that the lower bound stability underestimates the stability width at large values of  $a_3$  as anticipated,<sup>[97]</sup> but also underestimates the width at small values of  $a_3$ . This is supported by the data in Table 5.4.4, which provides  $a_3$ , the height of the point from the horizontal axis, and  $a_2 - a_1$ , which is proportional to the distance from the central symmetry axis. Therefore this  $a_2 - a_1$  can be used as a measure of stability. It indicates that molecular-like systems, i.e.,  $m_1, m_2 \ll m_3$ , are less affected by the mass difference between the like-charged particles as the system remains stable when  $m_2 \ll m_1$ , than are atomic-like systems,  $m_1, m_2 \ll m_3$ .

### 5.4.4. Comparison to Literature: The Existence of $e^+e^-p^+$

$e^+e^-p^+$  doesn't exist, or at least has not been observed in nature, and theory supports it as unstable.<sup>[115–117,120]</sup> This does not mean that it is unimportant. For example impacting a positron into a hydrogen atom causes its ionisation into a proton and Positronium.<sup>[142–146]</sup> This occurs naturally in weakly coupled plasma, making it important in understanding plasma and positrons in astrological processes.<sup>[144]</sup> For our purposes however, confirming its instability, serves as a useful check point for validating our methodology. Method 1 can not definitively calculate  $e^+e^-p^+$  as unstable instead placing it outside the stability domain where it is known to be stable, whereas Method 2 definitively predicts it as unstable as it calculates  $m_{cr}$  accurately to 2 s.f. to 3 s.f..

In Table 5.4.5 we give the critical mass of a particle required to bind with a hydrogen atom to be  $\geq 2.20(8) m_e$ . This, as in previous work,<sup>[115–117,120]</sup> disproves the existence of a positron ( $e^+$ ) binding to a hydrogen. To compare this critical mass to literature Rotenberg and Stein<sup>[115]</sup> calculated an upper bound for the critical mass as being  $m_{cr} \leq 2.20 m_e$  and more recently Rebane *et al.*<sup>[120]</sup> have calculated a

Property	Value
System	$p^+ m_{cr}^+ e^-$
$m_{cr}$	2.20(8)
Energy	-0.499 727 840
$E_{th}$	-0.499 727 840
$\langle \hat{T} \rangle$	0.499 727 839
$\langle \hat{V} \rangle$	-0.999 455 678

**Table 5.4.5.:** Our calculation of the critical mass of  $p^+ m_{cr}^+ e^-$  Using an optimised AC wavefunction with 8436 terms.

lower bound  $m_{cr} \geq 2.20(3) m_e$ . We can see that our results are right on par with the critical masses determined by other researchers.

## 5.5. Summary & Conclusions

The stability of a range of symmetric unit charge systems have been calculated to high accuracy and used to predict the width of the stability band. This was done in terms of the fractional additional binding,  $g$ , as a function of the reciprocal mass of the uniquely charged particle,  $a_3$ , using the method of Martin *et al.*<sup>[97]</sup> A functional form of  $g(a_3)$  is obtained by fitting to these data and is used to determine the minimum value of  $g(a_3)$ , the point of least relative excess binding due to a third particle. Mátyus *et al.*<sup>[100]</sup> discuss the emergence of molecular structure in three-body systems in terms of the density at the centre of mass and calculate that the transition from atom-like to molecule-like behaviour takes place within the interval  $0.4 < \frac{m_1}{m_3} < 0.8$ . The results presented here indicate that the transition from atom-like to molecule-like behaviour occurs when the system has the least energetic gain by association with the third particle.

The upper bound of the particle mass, corresponding to the exact threshold energy for a given two-body system, was calculated using the variational principle.<sup>[120]</sup> The generalised eigenvalue equation was solved using a series solution method for the three-particle system in translation-free internal coordinates using a wave function expanded in a triple orthogonal set of Laguerre polynomial functions with three non-linear variational parameters. This work demonstrates the utility of the series solution method, but indicates that the wavefunction, so successful in energy cal-

## 5. The Stability Domain for S-States of Unit Charge Coulomb 3-Body Systems

culations, does not adequately capture the physics of the variational mass problem, resulting in very slow convergence. However this is also true with others work<sup>[120]</sup> and seems to be a property of calculating critical mass variationally. The “exact” stability boundary of Method 2, with a precision of (2 to 3) s.f., was used to compare with the lower bound to stability recently calculated using Method 1. The shape and increased region of stability was discussed and the latter attributed to the important anti-symmetric terms in the Hamiltonian. Finally, a functional fit to the data in a judicious choice of mass coordinates provided a simple analytical expression for the calculation of the critical mass of a third particle required for stable binding to any two-body system.

With regards to future work, the surface of this work has only just been tapped upon. Perhaps of particular interest are the critical masses of the excited states of three-body systems. It is possible using Method 2 to generate a complete stability boundary for an excited state with little adaptation. In this boundary, not all symmetric systems are stable as it is known that  $H^-$  has no excited state.<sup>[139]</sup> Another avenue that could also be combined with an excited states extension would be non-unit charge systems. A new stability zone could be found for any combination of charges.

There also exist other ways of analysing our data on the ternary plot. Our current boundary shows the line at which  $E_0 = E_{th}$ . There is no reason not to plot the boundary where  $E_0 = 1.1E_{th}$  or any other factor. Doing this would describe a contour plot of the energy surface for all unit charge, three-body systems. Furthermore the function that plots this surface exactly would be the exact solution of the energy of three-body systems as a functions of mass for unit charge. Although we could not find that function using these contours, we can investigate how it behaves as in where it peaks, where its minima are and what is its general shape on the ternary diagram. This information could be used in a variety of ways including providing information as to the nature of an appropriate wavefunction for future work.

Work performed in this chapter has been published in the Journal of Chemical Physics.<sup>[125,136]</sup>

# 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

## 6.1. Introduction

In addition to the mass of the particles (Chapter 5), charge also influences the stability of a system consisting of three particles, under Coulomb interactions. Charge and mass constitute two key factors governing stability of a three-body system, whilst the third is the state the system is in. Three-body systems are again ideal for studying this stability. As discussed in Chapter 5, the energy of the lowest continuum threshold of a three-body system can be calculated from the two-body energy, which is known exactly for the non-relativistic time independent case as explained in Eq. (5.2.1). Three-body systems are relatively small compared to those found in conventional computational chemistry. However this allows for very accurate calculations of the energy and the wavefunction as described in Chapter 4. These calculations include non-adiabatic and correlation effects associated with the particles motion. In the present chapter the role of charge on the stability of two electron atoms and ions is considered.

The role of nuclear charge is shown in this chapter to have an effect on how electrons behave in terms of their spatial separation, with high charge causing the two electrons to have relatively small separations, and low charge causing them to separate. At a sufficiently low charge one of the electrons becomes unbound and the other behaves together with the nucleus like a two-body system. In such a spatial separation of electrons, these will be referred to as the inner and the outer

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

electron and calculations will be performed using a method not previously applied to wavefunctions with explicit  $r_3$  interactions, finite mass or arbitrary charges.

The studies in this chapter can be used to describe the effect of electron shielding. On reduction of the nuclear charge,  $Z_3$ , until one electron becomes unbound, this unbound electron is lost because the attraction of the nucleus has been shielded by the other electron, leaving only the repulsive force of the other electron. This electron is unbound when  $E \leq E_{\text{th}}$  ( $E$  is the ground state energy of a three-body system and  $E_{\text{th}}$  is the lowest continuum threshold energy of the same system). When this occurs, addition of an electron to a two-body system causes no gain in energy. However the work of Estienne *et al.*<sup>[147]</sup> suggests a shape resonance below critical nuclear charge ( $Z_{\text{cr}}$ ) where the electron remains localised to the nucleus.

Charge is an important factor for the effective computation of chemistry. It dictates the attractive and repulsive forces that make up the world of atoms and indeed molecules, and whilst some special cases of masses can be approximated as infinity, charge can not be. Of particular interest to a chemist is the nuclear charge and the effects of screening of that charge. Partial charges are not an unfamiliar topic for a chemist, with phenomena such as screening reducing the effective nuclear charge. Therefore the methodology employed with complete control of charge can study interesting effects caused by different charges directly as the computational approach allows the inclusion of non-integer charges.

The concept of critical charge has been discussed by various authors over the years, more recently there has been debate over the correct value for two electron atoms with various methods calculating different results. For example, the earlier work by authors such as Baker<sup>[148]</sup> are disputed by later work by Turbiner *et al.*<sup>[149]</sup> and by Zamastil.<sup>[150]</sup> Baker performed variational calculations on the  $\frac{1}{Z_3}$  expansion to determine an accurate  $Z_{\text{cr}}$  value. This critical charge was calculated with perturbation theory with 400 terms in the expansion. Guevara and Turbiner calculated the critical charge as  $0.91085e$  and Zamastil calculated it as  $0.9021e$ . These numbers are significantly different from each other whilst claiming to be accurate to the number of digits reported. To resolve this contradiction Estienne *et al.*<sup>[147]</sup> calculated a definitive  $Z_{\text{cr}}$  as  $0.911\,028\,224\,077\,255\,73e$  resolving contradiction in literature, and supporting the earlier work by Baker *et al.* In addition Estienne explored the elec-

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

tronic structure and behaviour of the outer electron as  $Z_3 \rightarrow Z_{\text{cr}}$ . This was done using the Hellman-Feynman theorem and  $\frac{1}{Z_3}$  expansion. This  $Z_{\text{cr}}$  value was calculated using the  $\frac{1}{Z_3}$  expansion as in previous work. The work of Estienne *et al.* was performed using a very accurate variational double and triple basis set method as described in the work by Drake *et al.* [89,113]

In this work we developed, programmed and implemented a method for direct computation of the critical charge. This method is variational and in contrast to all previous work, calculates the critical charge in a single computation for any given set of masses. All other methods by the authors noted above required many calculations of energy as a function of nuclear charge. In this work we consider finite mass, critical nuclear charges, some of which have not been previously considered. We also look into the properties of the critical charge system investigating its characteristics. These properties and this method are unique and have not been studied or implemented elsewhere.

**Inner and Outer Electrons** In this chapter the characteristics of systems with critical nuclear charge are studied. One such characteristic that was considered interesting was the individual properties of each electron for the purpose of studying electron detachment. To accomplish this, the electrons were separated into an inner and an outer component, a technique not widely used but useful for calculating certain properties such as effective nuclear charges. Hylleraas [63,64,151] first developed the variational method of including the  $r_3$  coordinate explicitly ( $r_3$  is sometimes referred to as  $r_{12}$  in the literature), developing a technique used by a variety of authors including this one. Hylleraas studied a number of concepts, properties and phenomena with this method including the effective charge that an outer and an inner electron experiences. To do this Hylleraas broke the integral domain of Hylleraas coordinates into an inner and an outer component. These domains form the base for the inner and outer electron properties, which will be discussed in this chapter. In addition, with the He wavefunction, Hylleraas introduced the concept of anti shielding, where the outer electron forces the inner electron closer to the nucleus causing an apparent increase in effective charge to values greater than that of the nuclear charge. As will be shown later this effect is also observed in this work. Koga *et al.* [152–159] has done the most in-depth work on the separate distributions of inner

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

and outer electrons. Unlike in earlier work, that focused on effective nuclear charge, in which inner and outer electrons are tools for probing certain quantities, Koga *et al.*<sup>[152–159]</sup> studied the inner and outer electrons themselves. These include inner and outer electron properties such as their radial distances ( $\langle r_i^n \rangle$ ) and their probability densities. Koga showed various relationships between these properties that we will explain in depth in Section 6.2.2. The work of Koga *et al.* was performed with the multi-configurational Hartree-Fock (MCHF) method, a technique that calculates energies in a more successful manner than traditional Hartree-Fock (HF). For example the energy for He is calculated as  $-2.903\,033\,E_h$ <sup>[160,161]</sup> as opposed to the true value of  $-2.903\,724\,E_h$ . Koga *et al.* has studied relatively large systems, such as atoms with  $d$  and  $f$  shells, and the effects inner electrons have on these outer shell electrons.

This work explicitly investigates the properties of the inner and outer electrons for atomic three-body systems. The inner and outer electrons are used to investigate in particular the “detachment” of an electron as the charge is decreased to and below the critical nuclear charge. This work is also the first to calculate the inner and outer electron properties with explicit inclusion of  $r_3$  and electron correlation effect as well as the first to include the correlated motion effect.

This chapter aims to introduce a novel, effective and efficient method developed to directly calculate critical charge. This method is then used to calculate the critical charge of both infinite and finite mass two-electron atoms to establish the minimum charge required to hold two electrons and have at least one bound state below the continuum threshold. The electronic structure is also revealed using a technique that separates inner and outer electrons by applying various operators to these inner and outer electrons. This shows the effect of nuclear charge on the electron behaviour of two fully correlated electrons.

## 6.2. Method

Chapter 5 demonstrated the stability of three-body systems as a function of mass but with fixed charge. This chapter determines the stability of three-body systems as a function of nuclear charge for two electron systems. A stable three-body system



## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

is defined as having at least one bound state more stable than the lowest continuum threshold, which in the present chapter corresponds to electron detachment.

### 6.2.1. Critical Charge

In this chapter a novel approach to calculating the minimum charge required by a nucleus to bind two electrons is presented. This will be referred to as the critical nuclear charge ( $Z_{\text{cr}}$ ). Within this chapter, any reference to the critical charge will be reference to the critical nuclear charge of the nucleus only. Here this is the critical nuclear charge of  $Z_3$ . Unlike the other methods mentioned above this approach calculates the critical nuclear charge as an eigenvalue, in much the same way as was calculated the critical mass ( $m_{\text{cr}}$ ) in Chapter 5 under “Method 2”. This critical nuclear charge method is variational, bounded from below, and an exact wavefunction calculates this charge exactly. The trial wavefunction described in detail in Section 4.2.8 on page 105 was used to variationally determine the critical charge. In Section 6.4.2 it will be shown that the rate of convergence of the critical nuclear charge is relatively rapid in stark contrast to the convergence rate of the critical mass. The rate of convergence for the critical nuclear charge was found to be much more comparable to the rate of convergence of the energy than that of the mass.

Rebane and Kuzminskii<sup>[120]</sup> showed that the lowest eigenvalue  $\lambda_0$  corresponds to  $\min K(\phi)$  where

$$K(\phi) = \frac{\int \phi^* \hat{P} \phi dV}{\int \phi^* \hat{Q} \phi dV} \quad (6.2.1)$$

Here  $\hat{P}$  is a self-consistent operator bounded from below and  $\hat{Q}$  is a positive definite, self-conjugate operator.<sup>[120]</sup> The computational method employed in this work has to have  $\hat{Q}$  as a positive definite operator or else the Cholesky decomposition as described in Chapter 3 fails to resolve with a fatal computational error. The eigenvalues  $\lambda_j$  are the extremals of  $K$ , satisfying the equation

$$\hat{P}\psi_j = \lambda_j \hat{Q}\psi_j, \quad j = 0, 1, 2, \dots \quad (6.2.2)$$

which arises from the requirement that the first variation vanishes, i.e.  $\delta K \equiv 0$ .<sup>[120]</sup>

This algorithm can be applied to the calculation of the threshold value of the

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

nuclear charge. The Schrödinger equation in atomic units for a two-electron atom of the form  $\{e^-e^-m_3^{Z_3+}\}$ , where  $m_3$  is the mass of the nucleus and  $Z_3$  is the nuclear charge, is as follows ( $m_3$  and  $Z_3$  are used to be consistent with the numbering schemes in other chapters):

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{Z_3}{r_1} - \frac{Z_3}{r_2} + \frac{1}{r_3}\right)\psi = E\psi \quad (6.2.3)$$

where  $r_1$  and  $r_2$  are the distances of the first and second electron from the nucleus, respectively, and  $r_3$  is their mutual separation, as described in Section 4.2.1 on page 89. A scaling of the coordinates  $r_i$  by  $Z_3$  results in the following  $Z$ -scaled Schrödinger equation:

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z_3} \frac{1}{r_3}\right)\psi = \frac{E}{Z_3^2}\psi \quad (6.2.4)$$

The purpose of such a scaling is to facilitate the rearrangement to make  $Z_3$  the eigenvalue in the following steps. At the stability threshold (the beginning of the essential spectrum of the operator), the wavefunction satisfies Eq. (6.2.4) with at the threshold  $E \equiv E_{th}$ . This  $E_{th}$  is known analytically as follows (the general form for  $E_{th}$  is given in Section 5.2 on page 136):

$$E \equiv E_{th} = -\frac{Z_3^2}{2} \frac{1}{1 + m_3^{-1}} \quad (6.2.5)$$

Where  $E_{th}$  is the ground state hydrogen-like, electron-nucleus, two-particle system that corresponds to the energy of the lowest dissociation threshold, which in this case is electron detachment. Therefore,

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z_3} \frac{1}{r_3}\right)\psi = \frac{E_{th}}{Z_3^2}\psi \quad (6.2.6)$$

Rearranging Eq. (6.2.6) gives a particular case of Eq. (6.2.2), giving the operators of  $\hat{P}$  and  $\hat{Q}$  for the critical charge.

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

First substitute Eq. (6.2.5) into Eq. (6.2.6) and simplify

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z_3 r_3}\right)\psi = -\frac{Z_3^2 \frac{1}{1+m_3^{-1}}}{Z_3^2}\psi \quad (6.2.7)$$

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z_3 r_3}\right)\psi = -\frac{Z_3^2}{2(1+m_3^{-1})Z_3^2}\psi \quad (6.2.8)$$

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{Z_3 r_3}\right)\psi = -\frac{1}{2(1+m_3^{-1})}\psi \quad (6.2.9)$$

Then rearrange to make all terms with  $Z_3$  on the right hand side, and all else on the left hand side.

$$\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{2(1+m_3^{-1})}\right)\psi = -\frac{1}{Z_3 r_3}\psi \quad (6.2.10)$$

When  $E = E_{th}$  then  $Z_3 = Z_{cr}$ . Giving the critical charge operators.

$$\underbrace{\left(-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{2(1+m_3^{-1})}\right)}_{\hat{P}}\psi = \underbrace{-\frac{1}{Z_{cr}}}_{\text{Eigenvalue}} \underbrace{\frac{1}{r_3}}_{\hat{Q}}\psi \quad (6.2.11)$$

$$\hat{P} = -\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{1}{2m_3}\nabla_3^2 - \frac{1}{r_1} - \frac{1}{r_2} + \frac{1}{2(1+m_3^{-1})} \quad (6.2.12)$$

$$\hat{Q} = \frac{1}{r_3} \quad (6.2.13)$$

These definitions of the operators satisfy the requirement that  $\hat{P}$  is bounded from below (c.f. the standard assumption that the Hamiltonian operator is bounded below in the variational principle<sup>[28]</sup>) and that  $\hat{Q}$  is a positive, definite, self-conjugate operator.<sup>[120]</sup> The lowest eigenvalue (which we label as  $\lambda_0$ ) corresponds to the threshold charge of the nucleus, i.e.,

$$\lambda_0 = -\frac{1}{Z_{cr}} \quad (6.2.14)$$

The higher eigenstates (i.e.  $\lambda_1, \lambda_2, \lambda_3, \dots$ ) correspond to the critical nuclear charge

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

of higher excited states. However the excited states are not the focus of this chapter, but with the program developed in this work such states are calculable and will be considered in future work.

The value of  $Z_{\text{cr}}$  corresponds to the nuclear charge for which the system is still stable. In the limit when the trial wavefunction tends to the exact solution of Eq. (6.2.6), this value tends to the exact value of the critical nuclear charge ( $Z_{\text{cr}}$ ). The generalised eigenvalue problem, Eq. (6.2.2), with the operators as defined in Eqs. (6.2.12) and (6.2.13) is solved to give the critical nuclear charge of a three-body system of the form  $\{e^-e^-m_3^{Z_3+}\}$  where  $m_3$  is the mass of the nucleus and  $Z_3(= Z_{\text{cr}})$  is the critical nuclear charge for binding. The lowest eigenvalue, corresponding to the singlet ground state, is acquired using the symmetric trial wavefunction as shown in Eq. (6.2.15) (Described in more detail in Section 4.2.9 on page 107). This wavefunction uses Laguerre polynomials of the form  $L_n(x)$  (Described in more detail in Section 4.2.8 on page 105). Perimetric coordinates are used and the problem was solved using the series solution method (with 57 terms) which allows for efficient calculation by avoiding integration. The wavefunction has 2 non-linear variational parameters  $A$  and  $C$  for reasons discussed in Chapter 4 meaning that the non-linear variational parameter  $B$  equals  $A$ . The symmetric wavefunction corresponds to the singlet ground state and is symmetric with respect to exchange of the  $r_1$  and  $r_2$  coordinate.

$$\begin{aligned} \psi(z_1, z_2, z_3) = e^{-\frac{1}{2}(Az_1+Bz_2+Cz_3)} \sum_{l,m,n} A(l, m, n) (L_l(Az_1) L_m(Bz_2) L_n(Cz_3) \\ + (\delta_{l,m} - 1) L_l(Az_2) L_m(Bz_1) L_n(Cz_3)) \end{aligned} \quad (6.2.15)$$

In this work, determinants of order 1078, 2856 and 4389 are considered, which correspond to wavefunctions represented by complete polynomials of order  $\omega = 21$ , 30 and 35, respectively, in the scaled, perimetric coordinates, in which  $l + m + n \leq \omega$  and the numbering takes advantage of the preserved symmetry.<sup>[71,73]</sup> These are referred to as Pekeris shells as described in Section 4.2.10 on page 109.

As discussed in Chapter 4 for the symmetric case, the constraint  $A = B$  is imposed to take advantage of the orthogonal<sup>[141]</sup> character of the wave function. However  $C$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

is allowed to vary independently to allow for an explicit dependence on  $r_3$  in the exponential behaviour at large  $r$ , which may be important at non-integer low  $Z_3$ . When  $C = A + B = 2A$  the exponent in the wave function models, in principle, the correct asymptotic behaviour of the solution of the Schrödinger equation for two-electron atoms at large  $r_1$  and  $r_2$ <sup>[63,64]</sup> (See Section 4.2.10 for a more in-depth discussion on the non-linear variational parameters, and Section 4.2.7 on page 100 for a detailed discussion of the “correct asymptotic” behaviour).

### 6.2.2. Distinguishing the Electrons: Inner and Outer Electrons

This chapter investigates the effect of decreasing the charge of a nucleus until an electron is no longer bound. To determine how the detachment occurs for a three-body system, as described by the wavefunction used here, a method of observing the “detachment” was developed. To do this it was necessary to distinguish between the two electrons, that, under conventional operators such as  $\langle r_1 \rangle$  and  $\langle r_2 \rangle$ , are identical due to symmetry and fermionic exchange. For this purpose the electron-nucleus coordinates of the wavefunction were separated into inner and outer distances. This is to say the electrons were spatial split into one electron that is closer to the nucleus ( $r_{in}$ ) and one that is further away ( $r_{out}$ ). Either particle 1 or particle 2 is considered as inner or outer at any moment as no distinction is made between the two particles. It was later discovered that this type of inner and outer electron separation had previously been investigated using multi-configurational Hartree-Fock (MCHF) type wavefunctions by Koga *et al.*<sup>[153,154,159,162]</sup>

To consider operators composed of  $r_{in}$  and  $r_{out}$  such as  $\langle r_{in} \rangle$  the following conditions were imposed upon the wavefunction:

$$r_i \leq r_j \quad \text{where } i = 1, 2 \text{ and } j = 1, 2 \text{ but } i \neq j \quad (6.2.16)$$

These conditions give us the ability to probe the spatial separation of two electrons under correlated motion. That is to say that any such correlated electrons would preferential prefer to separate in order to minimise electron repulsion. We define  $r_{in}$  as the inner electron distance from the nucleus, and  $r_{out}$  as the outer electron distance from the nucleus.

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

To enforce the conditions of Eq. (6.2.16) the Dirac delta function is used. The Dirac delta function is strictly speaking not a function but a distribution. However here it is referred to as a function to conform to convention.<sup>[163–168]</sup>

**Dirac Delta Function** The Dirac delta function  $\delta(x)$  was used to impose the condition within Eq. (6.2.16) upon the integral over the inter-particle distances. The Dirac delta is a distribution that is zero everywhere except at its origin.<sup>[85,163,167–169]</sup> The origin of the delta function is undefined except that the area under its curve is equal to 1. These “functions” therefore cause the wavefunction to vanish everywhere except at zero.<sup>[170]</sup> More specifically, the Dirac delta behaves as follows:<sup>[77,169–171]</sup> Note the definition of exclusive or ( $\oplus$ ) where  $A \oplus B$  is true when  $A$  or  $B$  is true but not when  $A$  and  $B$  are true.<sup>[172]</sup>

$$\int_a^b \delta(x) dx = \begin{cases} 1 & \text{if } 0 \in ]a, b[ \\ \frac{1}{2} & \text{if } a = 0 \oplus b = 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.2.17)$$

$$\int_a^b f(x) \delta(x) dx = \begin{cases} f(0) & \text{if } 0 \in ]a, b[ \\ \frac{f(0)}{2} & \text{if } a = 0 \oplus b = 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.2.18)$$

$$\int_a^b f(x) \delta(x - c) dx = \begin{cases} f(c) & \text{if } x - c = 0 \in ]a, b[ \\ \frac{f(c)}{2} & \text{if } a - c = 0 \oplus b - c = 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.2.19)$$

$$\int_a^b f(x) \delta(g(x)) dx = \begin{cases} \sum_{x_i} f(x_i) + \sum_{x_j} \frac{x_j}{2} & \begin{array}{l} \text{if roots of } g(x) \text{ exist in } [a, b] \\ \text{where } x_i \text{ is all roots of } g(x), x \in ]a, b[ \\ \text{and } x_j \text{ is all roots of } g(x), x \in \{a, b\} \end{array} \\ 0 & \text{otherwise} \end{cases} \quad (6.2.20)$$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

Equation (6.2.20) is the most complex but perhaps the most useful of the Dirac delta forms. In the most simple case when  $g(x) = x$  then Eq. (6.2.18) is obtained. When  $g(x) = x - c$  then Eq. (6.2.19) is obtained. More interesting cases of Dirac deltas exist when  $g(x) = 0$  for more than one value of  $x$ .

The following two cases describe how the Dirac delta behaves to give discrete points. This form is used to give us the probability densities of the wavefunction at certain distances along the coordinate and will later be used to calculate probability densities in Section 6.4.4. In the following examples of such cases  $a = 0$ , and  $b = \infty$ .

**Case 1** When  $g(x) = x^2 - 1$  then  $g(x) = 0$  when  $x = -1$  and  $x = 1$  in this case:

$$\int_0^\infty f(x)\delta(g(x)) dx = f(-1) + f(1) \quad (6.2.21)$$

**Case 2** When  $g(x) = \sin(x)$  then  $g(x) = 0$  when  $x = 0, \pi, 2\pi, \dots$ :

$$\int_0^\infty f(x)\delta(g(x)) dx = f(0) + f(\pi) + f(2\pi) + \dots \quad (6.2.22)$$

The following case evaluates to to a continuous range. This could also be represented as a Heaviside function. When  $g(x)$  gives ranges that are zero, the Dirac delta gives a continuous range as in the following case.

**Case 3** When  $g(x) = \begin{cases} 0 & \text{when } x > c \text{ and } x < d \text{ where } c > 0 \text{ and } d > c \\ 1 & \text{otherwise} \end{cases}$

$$\int_0^\infty f(x)\delta(g(x)) dx = \int_c^d f(x) dx \quad (6.2.23)$$

**Inner and Outer Operators** To calculate operators of the wavefunction such as the average distance the inner electron is from the nucleus, the inner and outer distances need to be defined. There are multiple definitions possible but here  $r_{in}$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

and  $r_{out}$  are defined as:

$$r_{in} = \min(r_1, r_2) \quad (6.2.24)$$

$$r_{in} = \frac{1}{2} (r_1 + r_2 - |r_1 - r_2|) \quad (6.2.25)$$

$$r_{out} = \max(r_1, r_2) \quad (6.2.26)$$

$$r_{out} = \frac{1}{2} (r_1 + r_2 + |r_1 - r_2|) \quad (6.2.27)$$

In this case the integral becomes:

$$\langle r_{in} \rangle = \int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* \frac{1}{2} (r_1 + r_2 - |r_1 - r_2|) \psi \, dr_3 \, dr_2 \, dr_1 \quad (6.2.28)$$

This form can be integrated by applying the following:

$$\int_{-\infty}^\infty |x| \, dx = \int_0^\infty x \, dx + \int_{-\infty}^0 -x \, dx \quad (6.2.29)$$

In practise these inner and outer operators are easier to apply than it would first seem as they can be evaluated simply as:

$$\langle r_{in} \rangle = \int_0^\infty \int_0^{r_1} \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_2 \psi \, dr_3 \, dr_2 \, dr_1 + \int_0^\infty \int_{r_1}^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_1 \psi \, dr_3 \, dr_2 \, dr_1 \quad (6.2.30)$$

To help understand these integrals and the inner and outer concepts we change the coordinate system. The following coordinate systems allow us to apply Eq. (6.2.29) to Eq. (6.2.28) where  $|r_1 - r_2|$  can be represented as  $|x|$ . There is already a coordinate system used for such purposes, referred to as Hylleraas coordinates, and these are discussed in the following discussion.

**Inner and Outer Coordinates: Hylleraas Coordinates** The discussion of inner and outer electrons has been in terms of  $r_i$  coordinates, however the calculations were performed in perimetric coordinates for the computational advantages associated with an independent coordinate system. Later in Section 6.3 the details of how this coordinate system is transformed, specifically for inner and outer electrons, to perimetric coordinates will be explained.

In the following discussion we use a coordinate system more natural to inner and



## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

outer electrons, namely the Hylleraas coordinates defined as:

$$\begin{aligned} s &= r_1 + r_2 \\ t &= r_1 - r_2 \\ u &= r_3 \end{aligned} \tag{6.2.31}$$

The integrals for including all of space in this coordinate system are as follows:

$$\int_0^\infty \int_{-\infty}^\infty \int_{|t|}^s f(s, t, u) du dt ds \tag{6.2.32}$$

In this coordinate system, the definition of  $r_{in}$  becomes much simpler:

$$r_{in} = \frac{1}{2}(s - |t|) \tag{6.2.33}$$

For  $r_{out}$ :

$$r_{out} = \frac{1}{2}(s + |t|) \tag{6.2.34}$$

For some function of  $r_{in}$  or  $r_{out}$  (that we shall call  $f(s, t, u)$ ) when integrated over all of space, the absolute function in the  $r_{in}$  or  $r_{out}$  is integrated by applying Eq. (6.2.29) which gives the following:

$$\int_0^\infty \int_0^\infty \int_{|t|}^s f(s, t, u)|_{|t|=t} du dt ds + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s f(s, t, u)|_{|t|=-t} du dt ds \tag{6.2.35}$$

This gives the  $r_1 \geq r_2$  sub-domain:

$$\int_0^\infty \int_0^\infty \int_{|t|}^s f(s, t, u) du dt ds \tag{6.2.36}$$

With the following as  $r_2 \geq r_1$  sub-domain:

$$\int_0^\infty \int_{-\infty}^0 \int_{|t|}^s f(s, t, u) du dt ds \tag{6.2.37}$$

From this we can immediately see that Eq. (6.2.36) covers precisely half of all of space, and that Eq. (6.2.37) also covers exactly half, or more specifically, the other half of all of space. Therefore when we apply an operator such as  $\delta(r_{in} - r)$  provided that we use both integrals Eqs. (6.2.36) and (6.2.37) are used all of space

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

is integrated over.

In a symmetric wavefunction the system is symmetric to exchange of the two electrons. From this we can take Eq. (6.2.37) and exchange  $r_1$  and  $r_2$ . When this is done we obtain Eq. (6.2.36), the following demonstrates:

Starting from Eq. (6.2.37)

$$\int_0^\infty \int_{-\infty}^0 \int_{|t|}^s f(s, t, u) du dt ds \quad (6.2.38)$$

Due to fermionic exchange  $r_1 \leftrightarrow r_2$ . In Hylleraas coordinates this is applied by transforming  $t$  to  $t'$  where  $t' = r_2 - r_1$

$$\int_0^\infty \int_{-\infty}^0 \int_{|t'|}^s f(s, t', u) du dt' ds \quad (6.2.39)$$

To transform back to  $t$ , the sign of  $t'$  is reversed such that  $t = -t'$ . As the sign of the integral coordinate is reversed the domain is also reversed giving Eq. (6.2.36)

$$\int_0^\infty \int_0^\infty \int_{|t|}^s f(s, t, u) du dt ds \quad (6.2.40)$$

Therefore with a system that is symmetric with respect to exchange, it is possible to calculate the full integral by taking only the first integral Eq. (6.2.36) and multiplying the result by 2. Indeed in Hylleraas's paper,<sup>[63,64,151]</sup> he simplified the calculation of  $\langle \hat{H} \rangle$  by taking only the  $r_1 \geq r_2$  domain and multiplying the resulting energy by 2.

**Inner and Outer Relations** The inner and outer electrons are related to the inter-particle distances through a set of relationships. Here these relations will be explained.

**Relationship 1** The first relationship states that the sum of the inner and outer electron radial distances from the nucleus gives the sum of the  $r_1$  and  $r_2$  distances.

$$\langle r_{in} \rangle + \langle r_{out} \rangle = \langle r_1 \rangle + \langle r_2 \rangle \quad (6.2.41)$$

This relation is true because to calculate the inner and outer radial distances

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

the integral ranges used to calculate  $\langle r_1 \rangle$  and  $\langle r_2 \rangle$  are split in half. Where the inner half of  $\langle r_1 \rangle$  and  $\langle r_2 \rangle$  are given to  $\langle r_{in} \rangle$  and  $\langle r_{out} \rangle$  is assigned the outer half.

Since an integral range  $\int_a^b f(x) dx$  can be split to the following  $\int_a^c f(x) dx + \int_c^b f(x) dx$ ,  $c \in ]a, b[$  and evaluate to the same value. Then adding inner and outer distances completes the integral range in much the same manner and sums to give  $\langle r_1 \rangle + \langle r_2 \rangle$ .

This can be seen mathematically for a generic operator  $A(r_i)$  in Hylleraas coordinates as:

$$\langle A(r_1) \rangle = \int_0^\infty \int_{-\infty}^\infty \int_{|t|}^s \psi^* A(r_1) \psi du dt ds \quad (6.2.42)$$

This can be split into the following two integrals

$$\begin{aligned} \langle A(r_1) \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_1) \psi du dt ds \\ &+ \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_1) \psi du dt ds \end{aligned} \quad (6.2.43)$$

$\langle A(r_2) \rangle$  similarly splits

$$\begin{aligned} \langle A(r_2) \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_2) \psi du dt ds \\ &+ \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_2) \psi du dt ds \end{aligned} \quad (6.2.44)$$

The inner and outer equivalents simply take different combinations of these

$$\begin{aligned} \langle A(r_{in}) \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_2) \psi du dt ds \\ &+ \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_1) \psi du dt ds \end{aligned} \quad (6.2.45)$$

$$\begin{aligned} \langle A(r_{out}) \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_1) \psi du dt ds \\ &+ \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_2) \psi du dt ds \end{aligned} \quad (6.2.46)$$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

Summing  $r_{in} + r_{out}$  gives the following

$$\begin{aligned}\langle A(r_{in}) \rangle + \langle A(r_{out}) \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_2) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_1) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_1) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_2) \psi \, du \, dt \, ds\end{aligned}\tag{6.2.47}$$

This trivially rearranges to

$$\begin{aligned}\langle A(r_{in}) \rangle + \langle A(r_{out}) \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_1) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_1) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* A(r_2) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* A(r_2) \psi \, du \, dt \, ds\end{aligned}\tag{6.2.48}$$

Finally substituting in Eqs. (6.2.43) and (6.2.44) gives the relationship

$$\langle A(r_{in}) \rangle + \langle A(r_{out}) \rangle = \langle A(r_1) \rangle + \langle A(r_2) \rangle\tag{6.2.49}$$

**Relationship 2** The second relationship states that the inner and outer electron radial distances can be calculated from the  $\langle r_1 \rangle$  distance  $\pm$  the average distance between  $r_1$  and  $r_2$  ( $\langle |r_1 - r_2| \rangle$ ). Where the negative combination gives  $r_{in}$  and the positive combination gives  $r_{out}$ .

$$\langle r_{in} \rangle = \langle r_1 \rangle - \langle |r_1 - r_2| \rangle\tag{6.2.50}$$

$$\langle r_{out} \rangle = \langle r_1 \rangle + \langle |r_1 - r_2| \rangle\tag{6.2.51}$$

This relation only holds for symmetric systems where  $\langle r_1 \rangle = \langle r_2 \rangle$ , as this means that  $\langle r_{in} \rangle + \langle r_{out} \rangle = 2 \langle r_1 \rangle$  and that  $\frac{\langle r_{in} \rangle + \langle r_{out} \rangle}{2} = \langle r_1 \rangle$ . Under these conditions  $\langle r_1 \rangle$  lies half way between  $r_{in}$  and  $r_{out}$ . The average distance between  $r_1$  and  $r_2$  can be calculated using  $\langle |r_1 - r_2| \rangle$  this is equivalent to the average distance between  $r_{in}$  and  $r_{out}$ . Therefore half of this average distance can be used to get from  $\langle r_1 \rangle$  to  $\langle r_{in} \rangle$  or  $\langle r_{out} \rangle$ .

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

This can be seen mathematically as follows:

$$\begin{aligned}\langle r_1 \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds\end{aligned}\tag{6.2.52}$$

$$\begin{aligned}\langle r_{in} \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_2 \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds\end{aligned}\tag{6.2.53}$$

$$\langle |r_1 - r_2| \rangle = \int_0^\infty \int_{-\infty}^\infty \int_{|t|}^s \psi^* |r_1 - r_2| \psi \, du \, dt \, ds\tag{6.2.54}$$

$$\begin{aligned}\int_{-\infty}^\infty |x| \, dx &\text{ can be evaluated into the following} \\ \int_{-\infty}^\infty |x| \, dx &= \int_0^\infty x \, dx + \int_{-\infty}^0 -x \, dx\end{aligned}\tag{6.2.55}$$

And as  $|r_1 - r_2| = |t|$  then Eq. (6.2.54) becomes:

$$\begin{aligned}\langle |r_1 - r_2| \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* (r_1 - r_2) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* (r_2 - r_1) \psi \, du \, dt \, ds\end{aligned}\tag{6.2.56}$$

This is simplified in much the same way as in Eqs. (6.2.38) to (6.2.40) due to the

fermionic exchange of  $r_1 \leftrightarrow r_2$

$$\begin{aligned}\langle |r_1 - r_2| \rangle &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* (r_1 - r_2) \psi \, du \, dt \, ds \\ &\quad + \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* (r_1 - r_2) \psi \, du \, dt \, ds\end{aligned}\tag{6.2.57}$$

$$= 2 \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* (r_1 - r_2) \psi \, du \, dt \, ds\tag{6.2.58}$$

Using the sum rule of integration to give

$$\begin{aligned}\langle |r_1 - r_2| \rangle &= 2 \left( \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \right. \\ &\quad \left. - \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_2 \psi \, du \, dt \, ds \right)\end{aligned}\tag{6.2.59}$$

$$\begin{aligned}\frac{\langle |r_1 - r_2| \rangle}{2} &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \\ &\quad - \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_2 \psi \, du \, dt \, ds\end{aligned}\tag{6.2.60}$$

It follows that Eq. (6.2.52) subtract Eq. (6.2.60) gives

$$\begin{aligned}
 \langle r_1 \rangle - \frac{\langle |r_1 - r_2| \rangle}{2} &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \\
 &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \\
 &\quad - \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \\
 &\quad + \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_2 \psi \, du \, dt \, ds \tag{6.2.61}
 \end{aligned}$$

$$\begin{aligned}
 &= \int_0^\infty \int_0^\infty \int_{|t|}^s \psi^* r_2 \psi \, du \, dt \, ds \\
 &\quad + \int_0^\infty \int_{-\infty}^0 \int_{|t|}^s \psi^* r_1 \psi \, du \, dt \, ds \tag{6.2.62}
 \end{aligned}$$

Finally substituting in Eq. (6.2.53) gives the relationship

$$\langle r_1 \rangle - \frac{\langle |r_1 - r_2| \rangle}{2} = \langle r_{in} \rangle \tag{6.2.63}$$

### 6.3. Implementation

All computational methods for solving eigenproblems in this chapter including the novel method for calculating the critical nuclear charge used the efficient series solution method that can avoid integration and thus reduce computational cost. The repulsive interaction of the like-charged particles was included explicitly from the beginning, and all particles were kept in motion unless otherwise stated. Atomic units are used throughout unless otherwise stated.

The non-linear variational parameters  $(A, C)$  were optimised using the method bound optimisation by quadratic approximation (BOBYQA) as described in Section 3.2.1 on page 35. All optimisations were performed in double precision (16-digit) for systems that are converged, in terms of energy, to less than 12 s.f.. However quadruple precision (32-digit) was required for the accurate calculation of expectation values involving Dirac delta functions discussed in Section 6.2.2. The particle-electron mass ratio for the helium nucleus (taken as the alpha particle), proton, muon and tauon were taken directly from the latest CODATA.<sup>[18]</sup> For  $Z_3 \geq 3$  the experimental atomic masses,  $M(\text{amu})$ , for the most abundant isotope with  $Z_3 \geq 3$ , were taken from the Atomic mass evaluation 2012,<sup>[173]</sup> and converted to atomic units via the relationship  $M(\text{au}) = M(\text{amu}) \times m_u/m_e$ . The value of the unified atomic

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

Initial System	Nuclear mass, $m_3$ (a.u.)
$\text{Ps}^-$	1
$\text{Mu}^-$	206.768 284 3
$\text{Tau}^-$	3477.15
$^1\text{H}$	1836.152 672 45
$^4\text{He}$	7294.299 536 1
$^7\text{Li}^+$	12 786.3918
$^9\text{Be}^{2+}$	16 424.2047
$^{11}\text{B}^{3+}$	20 063.7360
$^{12}\text{C}^{4+}$	21 868.661 82
$^{14}\text{N}^{5+}$	25 519.0423
$^{16}\text{O}^{6+}$	29 148.9456
$^{19}\text{F}^{7+}$	34 622.9703
$^{20}\text{Ne}^{8+}$	36 433.9889
$^\infty\text{H}^-$	$\infty$

**Table 6.3.1.:** Atomic systems described in the chapter and their nuclear masses. All of these systems contain two electrons and a single nucleus of finite mass.  $\text{Ps}^-$ ,  $\text{Mu}^-$  and  $\text{Tau}^-$  are exotic species with nucleus's of  $e^+$ ,  $\mu^+$  and  $\tau^+$  respectively. The masses for these exotic nuclei and for the  $\text{H}^-$  and  $\text{He}$  nuclei were taken from CODATA 2012<sup>[18]</sup> and have been given in Appendix B. The nuclear masses for the other atoms were taken from the Atomic Mass Evaluation (AME) 2012<sup>[173]</sup> and the appropriate number of electrons removed.

mass constant is  $m_\mu = 1.660\,538\,921(73) \times 10^{-27}$  kg, and the mass of the electron  $m_e = 9.109\,382\,91(40) \times 10^{-31}$  kg.<sup>[18]</sup> The nuclear masses in atomic units were then obtained by removing the appropriate number of electrons, and are provided in Table 6.3.1. All the systems listed in Table 6.3.1 are stable in their ground state and can be used as initial systems to determine the boundary of the region of stability for each system of a given mass,  $m_3$ , as the charge  $Z_3$  decreases. Note that due to the mass scaling rule these results also determine analogous systems with the same particle-mass ratios.

**Implementation of the Dirac Delta** To implement the Dirac Delta functions given in this chapter, each integral with a Dirac delta had to be explicitly evaluated as the computer algebra package Maple, with its built in “Dirac” function was incapable of handling these correctly or efficiently. In most cases the built in function returned unevaluated, only succeeding with the simpler cases. Therefore the Dirac deltas were first simplified by hand. For example the integral representing the particle densities

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

along  $r_1$  is given by  $\langle \delta(r_1 - r) \rangle$ :

$$\langle \delta(r_1 - r) \rangle = \int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* \delta(r_1 - r) \psi \, dr_3 \, dr_2 \, dr_1 \quad (6.3.1)$$

By applying the Dirac delta definition in Eq. (6.2.19) this then reduces as follows:

$$\langle \delta(r_1 - r) \rangle = \int_0^\infty \int_{|r-r_2|}^{|r+r_2|} \psi^* \psi|_{r_1=r} \, dr_3 \, dr_2 \quad (6.3.2)$$

With this form Maple is then capable of evaluating the integral effectively.

### The Effects of Exchange on Implementation of the Inner and Outer Electrons

The inner and outer average radial distances are calculated using the following expression.

$$\begin{aligned} \langle r_{in} \rangle &= \int_0^\infty \int_0^{r_1} \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_2 \psi \, dr_3 \, dr_2 \, dr_1 \\ &+ \int_0^\infty \int_{r_1}^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_1 \psi \, dr_3 \, dr_2 \, dr_1 \end{aligned} \quad (6.3.3)$$

Equation (6.3.3) can also be written as follows as this also imposes the appropriate conditions (take note of the order of integration):

$$\begin{aligned} \langle r_{in} \rangle &= \int_0^\infty \int_0^{r_1} \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_2 \psi \, dr_3 \, dr_2 \, dr_1 \\ &+ \int_0^\infty \int_0^{r_2} \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_1 \psi \, dr_3 \, dr_1 \, dr_2 \end{aligned} \quad (6.3.4)$$

The form of the average inner electron-nucleus distance shown in Eq. (6.3.4) is useful, since in atomic systems containing two electrons, the wavefunction is symmetric with respect to interchange of  $r_1$  and  $r_2$  (See Section 4.2.9 on page 107). As a result, the first and second terms of Eq. (6.3.3) integrate to an identical value. As was explained in Section 6.2.2 this means that in principle only one half of the integral range need be evaluated for systems with 2 identical particles.

**The Probability Densities** This chapter is the first to discuss the systems in terms of their probability densities. These expectation values, (See Section 4.4.3 on page 124 for further explanations of expectation values) denoted by  $\langle \delta(r_i - r) \rangle$



## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

represent the probability of coordinate  $r_i$  having the distance  $r$ . These were obtained by applying a Dirac delta to the probability with the following integral.

$$\langle \delta(r_1 - r) \rangle = \int_0^\infty \int_0^\infty \int_{|r_1 - r_2|}^{|r_1 + r_2|} \psi^* \delta(r_1 - r) \psi dr_3 dr_2 dr_1 \quad (6.3.5)$$

Applying this Dirac delta gives the following equation that can then be evaluated in Maple:

$$\langle \delta(r_1 - r) \rangle = \int_0^\infty \int_{|r - r_2|}^{|r + r_2|} \psi^* \psi|_{r_1=r} dr_3 dr_2 \text{ for } r \neq 0 \quad (6.3.6)$$

A significant technical detail in the implementation of Eq. (6.3.6) lies in its integral ranges. This function now integrates over the range  $r_2$  and  $r_3$  only and  $r_1$  has become a single given value. This greatly simplified the evaluation of these expectation values and as a result, it became a simple matter to plot the probability density for many values of  $r$ . The case when  $r = 0$  is special because at this point  $r_3 = r_2$  and the second integral vanishes. This can be seen by inspecting the range  $\int_{|r - r_2|}^{|r + r_2|}$  for when  $r = 0$  this reduces to the trivial  $\int_{|r_2|}^{|r_2|}$ . Therefore  $\langle \delta(r_1) \rangle$  becomes:

$$\langle \delta(r_1) \rangle = \int_0^\infty \psi^* \psi|_{r_1=0, r_3=r_2} dr_2 \quad (6.3.7)$$

Equation (6.3.7) is simple to evaluate with the wavefunction used here and can be evaluated entirely with recursion relations and series solution. Therefore the special case of this probability density, at coalescence, is computed in the much faster C++ code (Section 3.2.2 on page 50) rather than within Maple (Section 3.3.2 on page 67).

**Extending to Perimetric coordinates** The wavefunction is in perimetric coordinates as these are easier to calculate as an independent set of coordinates ranging from 0 to  $\infty$ .

So far we have discussed the expectation values of probability densities and the inner and outer electrons in terms of  $r_i$  coordinates. However by converting these expectations values to scaled perimetric coordinates it was found to be more efficient in computation, being some 2 to 5 times faster, as the integral ranges remain simpler with no  $|r_1 - r_2|$  component. The relevant integrals of the expectations values in  $r_i$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

coordinates are as follows, which will then transform to scaled perimetric coordinates:

$$\langle \delta(r_1 - r) \rangle = \int_0^\infty \int_{|r-r_2|}^{|r-r_2|} \psi^* \psi|_{r_1=r} dr_3 dr_2 \quad (6.3.8)$$

$$\begin{aligned} \langle r_{in} \rangle &= \int_0^\infty \int_0^{r_1} \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_2 \psi dr_3 dr_2 dr_1 \\ &+ \int_0^\infty \int_{r_1}^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} \psi^* r_1 \psi dr_3 dr_2 dr_1 \end{aligned} \quad (6.3.9)$$

$$\langle \delta(r_{in} - r) \rangle = \int_r^\infty \int_{|r_1-r|}^{|r_1+r|} \psi^* \psi|_{r_2=r} dr_3 dr_1 + \int_r^\infty \int_{|r-r_2|}^{|r+r_2|} \psi^* \psi|_{r_1=r} dr_3 dr_2 \quad (6.3.10)$$

The scaled perimetric coordinates are applied

$$r_1 = \frac{v}{2B} + \frac{w}{2C} \quad (6.3.11)$$

$$r_2 = \frac{u}{2A} + \frac{w}{2C} \quad (6.3.12)$$

$$r_3 = \frac{u}{2A} + \frac{v}{2B} \quad (6.3.13)$$

Applying these to Eq. (6.3.8) we note that when

$$r_1 = r \quad (6.3.14)$$

$$\frac{v}{2B} + \frac{w}{2C} = r \quad (6.3.15)$$

This is first rearranged to make either  $v$  or  $w$  the subject. Arbitrarily,  $w$  was chosen.

$$w = 2C \left( r - \frac{v}{2B} \right) \quad (6.3.16)$$

Therefore we substitute for  $w$  instead of  $r_1$  (leaving integral ranges out for this step)

$$\int \int \psi^* \psi|_{w=2C(r-\frac{v}{2B})} dv dw \quad (6.3.17)$$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

The integral domains must cover all of space for their coordinate. For  $u$  this remains 0 to  $\infty$ , however, for  $v$ , we must consider the effect it has on  $w$ . According to Eq. (6.3.16) if  $v$  becomes larger than  $2Br$  then  $w$  would be negative.  $w$  must remain within its original domain of 0 to  $\infty$  therefore  $v$  is limited to the range 0 to  $2Br$ . This gives the integral for  $\langle \delta(r_1 - r) \rangle$

$$\langle \delta(r_1 - r) \rangle = \int_0^\infty \int_0^{2Br} \psi^* \psi|_{w=2C(r-\frac{v}{2B})} dv dw \quad (6.3.18)$$

A similar process is employed to calculate the other integrals.

$$r_1 \geq r_2 \quad (6.3.19)$$

$$\frac{v}{2B} + \frac{w}{2C} \geq \frac{u}{2A} + \frac{w}{2C} \quad (6.3.20)$$

$$\frac{v}{2B} \geq \frac{u}{2A} \quad (6.3.21)$$

$$v \geq \frac{Bu}{A} \quad (6.3.22)$$

$$\begin{aligned} \langle r_{in} \rangle &= \int_0^\infty \int_{\frac{Bu}{A}}^\infty \int_0^\infty \psi^* \left( \frac{u}{2A} + \frac{w}{2C} \right) \psi dw dv du \\ &\quad + \int_0^\infty \int_0^{\frac{Bu}{A}} \int_0^\infty \psi^* \left( \frac{v}{2B} + \frac{w}{2C} \right) \psi dw dv du \end{aligned} \quad (6.3.23)$$

$$\begin{aligned} \langle \delta(r_{in} - r) \rangle &= \int_0^{2Ar} \int_{\frac{Bu}{A}}^\infty \psi^* \psi|_{w=2C(r-\frac{u}{2A})} dv du \\ &\quad + \int_0^{2Br} \int_{\frac{Av}{B}}^\infty \psi^* \psi|_{w=2C(r-\frac{v}{2B})} du dv \end{aligned} \quad (6.3.24)$$

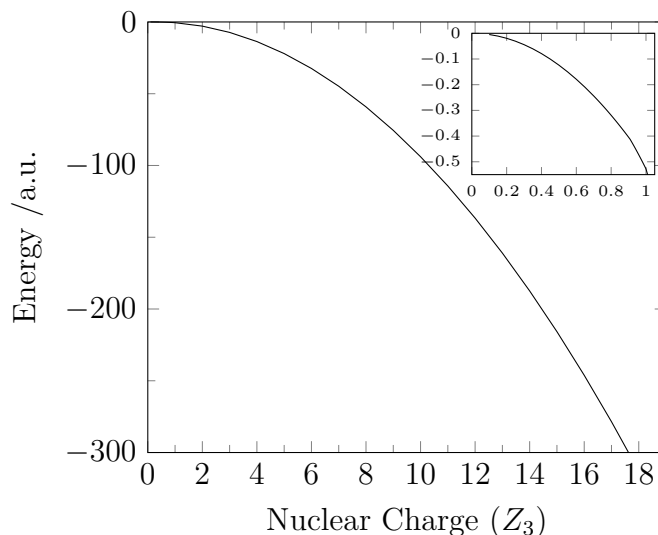
These integrals in scaled perimetric coordinates were implemented in Maple and used to calculate the various properties of the wavefunction. It is worth noting that here, the series solution was not applied and so these integrals were 2000 times slower compared to the other expectation values calculated in C++ such as  $\langle r_1 \rangle$ . For example, calculating  $\langle r_1 \rangle$  using Maple took 2.5 hours compared to 5 seconds in C++ for a 2856-term wavefunction.

## 6.4. Results and Discussion

### 6.4.1. Energy as a function of nuclear charge

The calculated energies for systems such as He and  $H^-$  are known to be of good quality, as described in Chapter 4 on page 87, and are in excellent agreement with the

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge



**Figure 6.4.1.:** Ground-state energy of  $\{e^-e^-m_3^{Z_3+}\}$ , with mass  $m_3 = \infty$ , as a function of  $Z_3$ .

most recent high accuracy values reported in the literature.<sup>[65,174]</sup> It is worth noting that the agreement between the finite masses used here, and those in the literature, is no greater than 8 s.f. for  $Z_3 \geq 3$ . The energies for systems with the nucleus in motion are less negative than those with a fixed nucleus. The energy difference is of the order of  $10^{-4}$  a.u. for  $H^-$  to  $Be^{2+}$  and  $10^{-3}$  a.u. for  $B^{3+}$  to  $Ne^{8+}$  and so increases from less than  $1 \text{ kJ mol}^{-1}$  to about  $7 \text{ kJ mol}^{-1}$ . The effect of nuclear motion on the total energy of the system becomes smaller as the nuclear mass increases, as expected (See Appendix C about concave functions of the energy for parameters entering the Hamiltonian linearly).

In Figure 6.4.1, the energy of the ground state of two-electron atoms, corresponding to the two-electron Hamiltonian with  $m_3 = \infty$ , is shown for a range of integer and non-integer values of  $Z_3$ . A general theorem exists stating that the energy is a concave function of any parameter entering the Hamiltonian linearly.<sup>[104]</sup> The ground state energy is therefore a concave function of the charge,  $Z_3$ . It is clear from Figure 6.4.1 that, as expected, the energy continues to be a continuous function of  $Z_3$  for non-integer low  $Z_3$  values.

### 6.4.2. Critical Charge required for binding

The critical nuclear charge required for binding was determined using the variational method described in Section 6.2.1 with a range of nuclear masses from 1 to infinity

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

Initial System	$Z_{\text{cr}}$ (a.u.)
$\text{Ps}^-$	0.921 802 4(4)
$\text{Mu}^-$	0.911 392 7(8)
$\text{Tau}^-$	0.911 050 1(6)
$^1\text{H}$	0.911 069 7(3)
$^4\text{He}$	0.911 038 6(9)
$^7\text{Li}^+$	0.911 034 2(0)
$^9\text{Be}^{2+}$	0.911 032 8(8)
$^{11}\text{B}^{3+}$	0.911 032 0(3)
$^{12}\text{C}^{4+}$	0.911 031 7(2)
$^{14}\text{N}^{5+}$	0.911 031 2(2)
$^{16}\text{O}^{6+}$	0.911 030 8(5)
$^{19}\text{F}^{7+}$	0.911 030 4(3)
$^{20}\text{Ne}^{8+}$	0.911 030 3(2)
$^\infty\text{H}^-$	0.911 028 2(3)

**Table 6.4.1.:** Bound on the critical nuclear charge,  $Z_{\text{cr}}$ , for two-electron systems, calculated using a 2856-term basis with two non-linear variational parameters; the digit given in parenthesis is not considered converged.

(Table 6.4.1). Additionally, to check this new method, the total energy for each  $\{e^-e^-m_3^{Z_3+}\}$  system was calculated using the standard Hamiltonian in Eq. (6.2.3), and compared with the threshold energy,  $E_{th} = -Z_3^2/2(1+m_3^{-1})$  for that particular  $Z_3$  to determine if the three-body system remained bound. Given that the variational energy calculations provide an upper bound to the true energy, the smallest value of  $Z_3$  for which the calculated energy lies below the electron detachment threshold is an upper bound to the exact value of the critical nuclear charge ( $Z_{\text{cr}}$ ). Similarly, the variational method for the direct determination of  $Z_3$  described in Section 6.2.1 provides an upper bound. Convergence behaviour was determined by evaluating the energy and critical  $Z_3$  as a function of basis set size. The convergence of  $Z_3$  near the boundary of stability was slower than the energy convergence (Table 6.4.2). It was found that both methods, the variational method for the direct determination for  $Z_{\text{cr}}$  and the energy comparison method, provided the same value for  $Z_{\text{cr}}$  to the reported accuracy for a given basis set size but the new method provided the value in a single calculation reducing the computational effort significantly.

Using the series solution method with a 2856-term wavefunction, it is found that  $Z_{\text{cr}} = 0.9110282$ , which is in excellent agreement with the most recent literature value of  $Z_{\text{cr}} = 0.91102822407725573(4)$  using a multiple basis set method with up

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

No. of Terms, $N$	Energy of $^\infty\text{He}$ (a.u.)	$^\infty Z_{\text{cr}}$ (a.u.)
1078	<b>-2.903 724 377 026 604</b>	<b>0.911 028 431 8</b>
2856	<b>-2.903 724 377 034 010</b>	<b>0.911 028 235 5</b>
4389	<b>-2.903 724 377 034 099</b>	<b>0.911 028 227 1</b>
8924	<b>-2.903 724 377 034 104</b>	<b>0.911 028 224 4</b>

**Table 6.4.2.:** Rate of convergence with basis set size (number of basis functions  $N$ ). Converged digits are in bold. The energy convergence of the helium atom is compared with the charge convergence for  $Z_{\text{cr}}$ . In both cases, the nuclear mass is taken as infinite.

to 2276 terms.<sup>[147]</sup> Additional converged digits are obtained by going to larger basis set sizes (Table 6.4.2). Additionally, the results presented here using a Laguerre polynomial-based single basis set method, further support the results obtained by Estienne *et al.*<sup>[147]</sup> regarding the contradictory values in the literature.

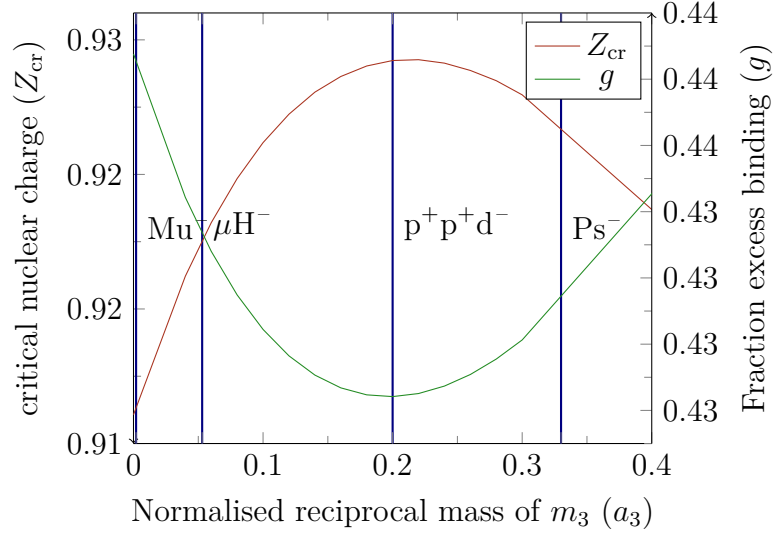
The minimum (critical) charge of the third particle/nucleus required to bind two-electrons is given in Table 6.4.1 starting from the stable systems  $\{e^-e^-m_3^{Z_3+}\}$  with finite nuclear masses,  $m_3$ , ranging from the mass of a positron = 1 to the mass of a neon nucleus = 36 433.9889. Rebane<sup>[126]</sup> has calculated the regions of unconditional stability and instability of Coulomb systems with varying particle charges for fixed ratios of the particles masses and reports values starting from  $\text{Ps}^-$  and  $\text{Mu}^-$  of 0.9296 and 0.9174, respectively. The results in Table 6.4.1 for these systems improve on these upper bounds (0.921 802 4 and 0.911 392 7, respectively) and are the most accurate reported to date. Furthermore, to our knowledge, this is the first time the upper bound on the critical value of the charge has been reported for a range of systems with heliogenic nuclear masses. The data reveal that as the mass decreases, a greater nuclear charge is required to ensure binding. The difference in nuclear charge for the helium isoelectronic sequence is of the order of  $4 \times 10^{-5}$ , but as the mass is reduced from that of a proton to that of a positron, the difference in the critical charge for binding is much greater at  $1 \times 10^{-2}$ . This is attributed in a physical sense to the increased nuclear motion of a light particle requiring a greater attractive nucleus-electron interaction to trap the second electron and overcome the electron-electron repulsion. In an energetic sense the  $\text{Ps}^-$  has less fractional excess binding energy ( $g$ ) (as discussed in Section 5.4.1 on page 153) and was therefore closer to the critical threshold to start with.

It would seem from these results that by reducing the mass of  $m_3$  relative to  $m_1$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

and  $m_2$  the critical charge rises. It was thought however that instead of always increasing it would again begin to diminish as  $\frac{m_3}{m_1}$  ratio becomes less than 1 and we enter the molecular domain. This is for the same reasons outlined in Chapter 5 and due to the quadratic nature of the fractional excess binding energy ( $g$ ) as a function of the  $m_3$  mass. Using the novel method for efficient calculation of  $Z_{\text{cr}}$ , it was possible to quickly calculate many critical nuclear charges for many systems with masses between  $\text{H}^-$  to  $\text{Ps}^-$  and investigate the effect  $m_3$  mass has on the critical charge. Even systems that do not correlate to known particles were included in order to explore the entire domain. As energy is a concave function of charge then the amount of charge needed to be lost to reach the critical point ( $E = E_{th}$ ) is proportional to the amount of fractional excess binding energy i.e. how close in terms of fractional excess binding energy the energy is to 0. Figure 6.4.2 shows a plot of  $Z_{\text{cr}}$  and the fractional excess binding energy of unit charge systems, demonstrating this relationship. The fractional excess binding energy values were calculated using the  $g$  function we published on this matter in J. Chem. Phys (2013)<sup>[125]</sup> this is also laid out in Section 5.4.1 on page 153. The masses in Figure 6.4.2 are given in normalised reciprocal mass ( $a_i$ ) where  $a_i = \frac{m_i^{-1}}{m_1^{-1} + m_2^{-1} + m_3^{-1}}$ ,  $i = 1, 2, 3$  as described in Section 5.2.1 on page 137. All of these new points were calculated with a relatively small wavefunction of 1078 terms. This smaller wavefunction, although not as accurate as the large 2856+ wavefunctions still gives converged  $Z_{\text{cr}}$  values to 4 s.f. and is therefore perfectly adequate for the scale of the figure.

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge



**Figure 6.4.2.:** The trend of critical nuclear charge and the fractional excess binding energy. The vertical lines correspond to physical symmetric systems.

### 6.4.3. Bound state properties as a function of nuclear charge

To determine the quality of the solutions obtained, various expectation values have been calculated using the best 2856-term wavefunction in each case. The properties presented in Table 6.4.3 include the expectation values of the inter-particle distances  $r_i$ , and various powers,  $r_i^{-1}$  and  $r_i^2$ , the two-particle Dirac delta functions,  $\delta(r_i)$ , and the two-particle cusps,  $\nu_i$ . In this table, the subscript 1 refers to the nucleus-electron interaction and the subscript 3 refers to the electron-electron interaction.

As discussed in Section 4.4.3 on page 123 these properties can determine the quality of the wavefunction. The extent to which the virial condition  $\langle \hat{V} \rangle = -2 \langle \hat{T} \rangle$  is satisfied provides a measure of the quality of the solution. It was found that for all systems, including at  $Z_{cr}$ , the factor  $\eta$  defined in Eq. (6.4.1) were calculated to be less than  $5 \times 10^{-11}$ , which is close to the exact value of zero.

$$\eta = \left| \frac{\langle \hat{V} \rangle}{\langle \hat{T} \rangle} + 2 \right| \quad (6.4.1)$$

Where  $\langle \hat{V} \rangle$  and  $\langle \hat{T} \rangle$  are the expectation values of the potential and kinetic energy, respectively.

Expectation values of the inter-particle coordinates for the systems considered in Table 6.4.1 are in very good agreement with the values reported by Frolov<sup>[111,175]</sup> for



	$^\infty\text{He}$	$^\infty\text{H}^-$	$^\infty Z_{\text{cr}}$	$^4\text{He}$	$^4 Z_{\text{cr}}$
$\langle r_1 \rangle$	0.929 472 294 87(4)	2.710 178 27(3)	4.146 663	0.929 607 915 01(2)	4.147 455
$\langle r_3 \rangle$	1.422 070 255 5(7)	4.412 694 4(8)	7.082 812	1.422 247 512 607(4)	7.084 170
$\langle r_1^{-1} \rangle$	1.688 316 800 7(0)	0.683 261 767 6(9)	0.578 109	1.688 076 584 66(8)	0.578 030
$\langle r_3^{-1} \rangle$	0.945 818 448(8)	0.311 021 502(3)	0.223 37(5)	0.945 697 223 219(6)	0.223 338
$\langle r_1^2 \rangle$	1.193 482 995 02(8)	11.913 699(4)	39.745 475	1.193 834 894 42(3)	39.762 644
$\langle r_3^2 \rangle$	2.516 439 312 856(8)	25.202 02(4)	81.234 426	2.517 061 842 48(4)	81.268 325
$\langle \delta(r_1) \rangle$	1.810 429(3)	0.164 552 8(5)	0.172 887	1.809 672(3)	0.172 814
$\langle \delta(r_3) \rangle$	0.106(34)	0.002 73(8)	0.001 618	0.106(30)	0.001 617
$\eta$	$6.48 \times 10^{-13}$	$1.28 \times 10^{-13}$	$1.25 \times 10^{-11}$	$1.23 \times 10^{-12}$	$4.32 \times 10^{-11}$
$\nu_1^{\text{a}}$	-1.999 999	-0.999 998	-0.91(0)	-1.999 72(3)	-0.910(8)
$\nu_3^{\text{b}}$	0.499 983	0.499 978	0.497 803	0.499 969	0.497 802

<sup>a</sup> The exact value of  $\nu_1$  for the infinite nuclear mass systems is  $-Z_3$ . For  $^4\text{He}$  the exact value is  $-1.999\,725$  and for  $^4 Z_{\text{cr}}$   $0.910\,913$ .

<sup>b</sup> The exact value of  $\nu_3$  is  $\frac{1}{2}$ .

**Table 6.4.3.:** Expectation values  $\langle X_i \rangle$  in atomic units, of some bound state properties for the ground states of He,  $\text{H}^-$  and  $Z_{\text{cr}}$ . The He and  $\text{H}^-$  data is reported to known accuracy by comparison with literature<sup>[111,175]</sup> (where the last digit in parenthesis is not necessarily converged) to gauge the accuracy of the values at  $Z_{\text{cr}}$ . The subscripts 1 and 3 designate the nucleus-electron and electron-electron interactions, respectively. Data was calculated using a 2856-term wavefunction. For comparison, the  $Z_{\text{cr}}$  expectation values obtained using the standard Hamiltonian in Eq. (6.2.3), rather than the  $Z$ -scaled Hamiltonian in Eq. (6.2.4), are reported.

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

the helium isoelectronic sequence; the He and  $H^-$  values are provided in Table 6.4.3. To judge the quality of the wavefunction at non-integer  $Z_3$  the value of  $\langle r_3^{-1} \rangle$  at  $Z_{cr}$  was compared with that reported in the literature for the infinite nuclear mass system. Estienne *et al.*<sup>[147]</sup> used the Hellman-Feynman theorem

$$\frac{dE}{d(1/Z_3)} = \int \psi^* \frac{\partial \hat{H}}{\partial (1/Z_3)} \psi dv = \left\langle \frac{1}{r_3} \right\rangle \quad (6.4.2)$$

where  $\hat{H}$  is the Hamiltonian of the system and  $\psi$  the wave function, to confirm the accuracy of their value calculated as the slope of a linear extrapolation of infinite nuclear mass energy against reciprocal nuclear charge  $Z_3$ . In this work, the calculated value of  $\langle r_3^{-1} \rangle$  at  $Z_{cr}$  using the  $Z$ -scaled Hamiltonian is 0.2451(9) which agrees well with their reported value of 0.245 189 063 9(1). However, a wavefunction calculated using a  $Z$ -scaled Hamiltonian will give  $Z$ -scaled expectation values. Therefore, rather than the  $Z$ -scaled Hamiltonian in Eq. (6.2.4), the true expectation value, as in that which could be measured experimentally, is calculated using the standard Hamiltonian (Equation (6.2.3)), which yields a value of  $\langle r_3^{-1} \rangle = 0.2233(7)$ . In practise the  $Z$ -scaled expectation value reported previously can be unscaled with the appropriate factor or an unscaled wavefunction can be obtained by recalculating the system and using the Hamiltonian of Eq. (6.2.3) and expectation values could then be calculated using this unscaled wavefunction.

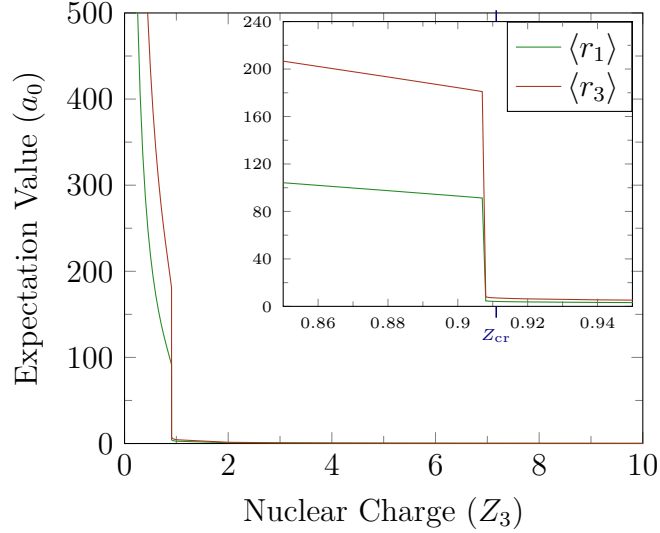
The two-body cusp ratios were determined using<sup>[105,111]</sup>:

$$\nu_i = \langle \hat{\nu}_i \rangle = \frac{\left\langle \delta(r_i) \frac{\partial}{\partial r_i} \right\rangle}{\langle \delta(r_i) \rangle} \quad (6.4.3)$$

The exact value of  $\nu_1$ , the nucleus-electron cusp, is  $-\frac{Z_3 m_3}{1+m_3}$ <sup>[105]</sup> which reduces to  $-Z_3$  for the infinite nuclear mass systems, and the exact value of  $\nu_3$ , the electron-electron cusp, is  $\frac{1}{2}$ . The values in Table 6.4.3 indicate that the cusp values at  $Z_{cr}$  are not quite as good as those for helium and the hydride ion but are quite reasonable.

Expectation values of the inter-particle coordinates  $\langle r_1 \rangle$  and  $\langle r_3 \rangle$  as a function of nuclear charge  $Z_3$  are shown in Figure 6.4.3. There is no appreciable difference between the figure for mass  $m_3 = \infty$  and when taking the finite mass equal to that of a proton, i.e.,  $M = 1836.152\,672\,45$  a.u., as the values differ only in the 4<sup>th</sup> or 5<sup>th</sup> s.f.,

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

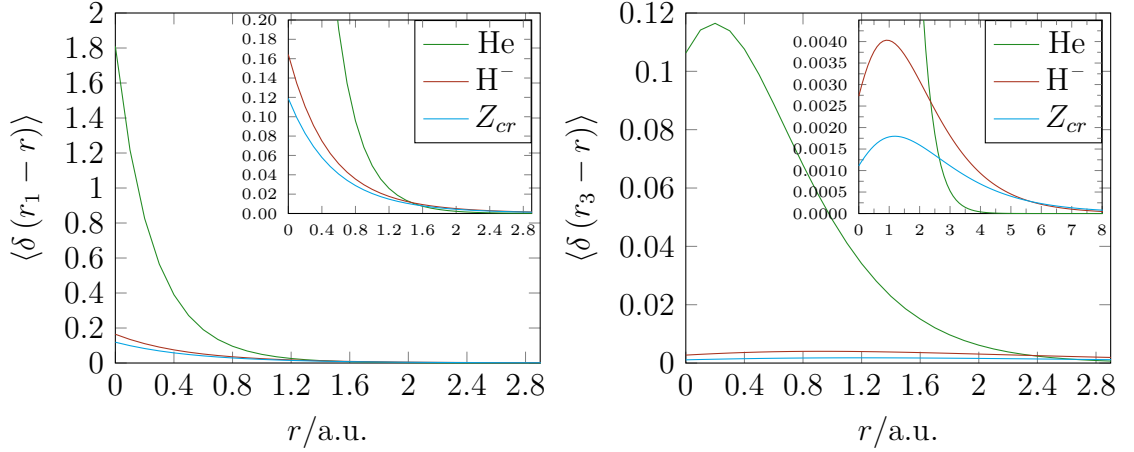


**Figure 6.4.3.:** Expectation values  $\langle r_1 \rangle$  and  $\langle r_3 \rangle$  with  $m_3 = m_p = 1836.152\,672\,45$ , as a function of nuclear charge,  $Z_3$ . The critical nuclear charge,  $Z_{cr}$ , is labelled in the inset. This figure is representative for all  $m_3 \geq m_p$ . Large values of  $r_i$  (i.e.  $> 100\,a_0$ ) are considered as  $\infty$  on the scale of three-body atoms and molecules.

therefore only the latter is shown. For  $Z_3 < 1$  (see inset of Figure 6.4.3), the electrons appear to remain weakly bound to just below  $Z_{cr}$  and then there is an abrupt jump to large values of  $\langle r_i \rangle$  as the nuclear charge is no longer sufficient to bind the electrons. No such jump is observed in the total energy (Figure 6.4.1) at this point. Figure 6.4.3 indicates that close to  $Z_{cr}$  (i.e., at  $Z_3 \approx 0.910$ ) the electrons remain localised, even though the total energy is above the lowest continuum threshold. Complete electron detachment occurs at  $Z_3 < 0.910$ . Estienne *et al.*<sup>[147]</sup> observed that their non-linear variational parameter describing the asymptotic behaviour of the outer electron does not tend to zero as  $Z_3 \rightarrow Z_{cr}$  as would happen if the outer electron moved to infinity at the critical point. They attributed localisation of the wavefunction at a finite distance from the nucleus below the critical point to the existence of resonances induced by the shape of the atomic potential. It would appear that these shape resonances, at  $0.9 < Z_3 < Z_{cr}$ , are detectable in the data presented in Figure 6.4.3, and that the behaviour is independent of the nuclear motion.

To further elucidate the electronic structure near the critical nuclear charge, the probability distribution for the electron-nucleus distance,  $r_1(= r_2)$  and for the

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge



**Figure 6.4.4.:** Probability distributions calculated using (a)  $\langle \delta(r_1 - r) \rangle$  and (b)  $\langle \delta(r_3 - r) \rangle$ . For  $^4\text{He}$  ( $Z_3 = 2$ ),  $^1\text{H}^-$  ( $Z_3 = 1$ ) and  $Z_{cr}$  for the finite nuclear mass systems where  $m_3 = m_p = 1836.152\,672\,45$  for  $Z_{cr}$ . These figures are indistinguishable from those with  $m_3 = \infty$ .

electron-electron distance  $r_3$  have been calculated using:

$$\langle \delta(r_i - r) \rangle = \int_0^\infty \int_0^\infty \int_{|r_1 - r_2|}^{|r_1 + r_2|} \psi^*(r_1, r_2, r_3) \delta(r_i - r) \psi(r_1, r_2, r_3) dr_1 dr_2 dr_3 \quad (6.4.4)$$

The probability distribution for the helium atom ( $Z_3 = 2$ ) is compared to that of the hydride ion ( $Z_3 = 1$ ) and the critical nuclear charge system ( $Z_3 = Z_{cr}$ ) in Figure 6.4.4. Again, there is no discernible difference between the data for the infinite nuclear mass systems and that for the finite nuclear mass systems, where the nuclear mass at  $Z_{cr}$  is taken as the proton mass. This is because changes due to nuclear motion appear in the 4<sup>th</sup> significant figure. The nucleus-electron probability density, often referred to as the single-particle density, and the electron-electron probability density, often referred to as the intracule density, show that all three systems have the same basic profile. The expected “cusp” condition of the nucleus-electron probability distribution is much less pronounced in the anionic systems, due to the reduced nuclear charge. A maximum occurs in the electron-electron distribution for all three systems but the density is greatly diminished and the maximum shifts to greater electron-electron separation for the anionic systems. Furthermore, as the charge decreases the probability distribution goes to zero much more slowly. It is clear from these figures that the excess negative charge dominates the interactions, resulting in the electron density becoming more diffuse, diminishing electron density close to the nucleus.

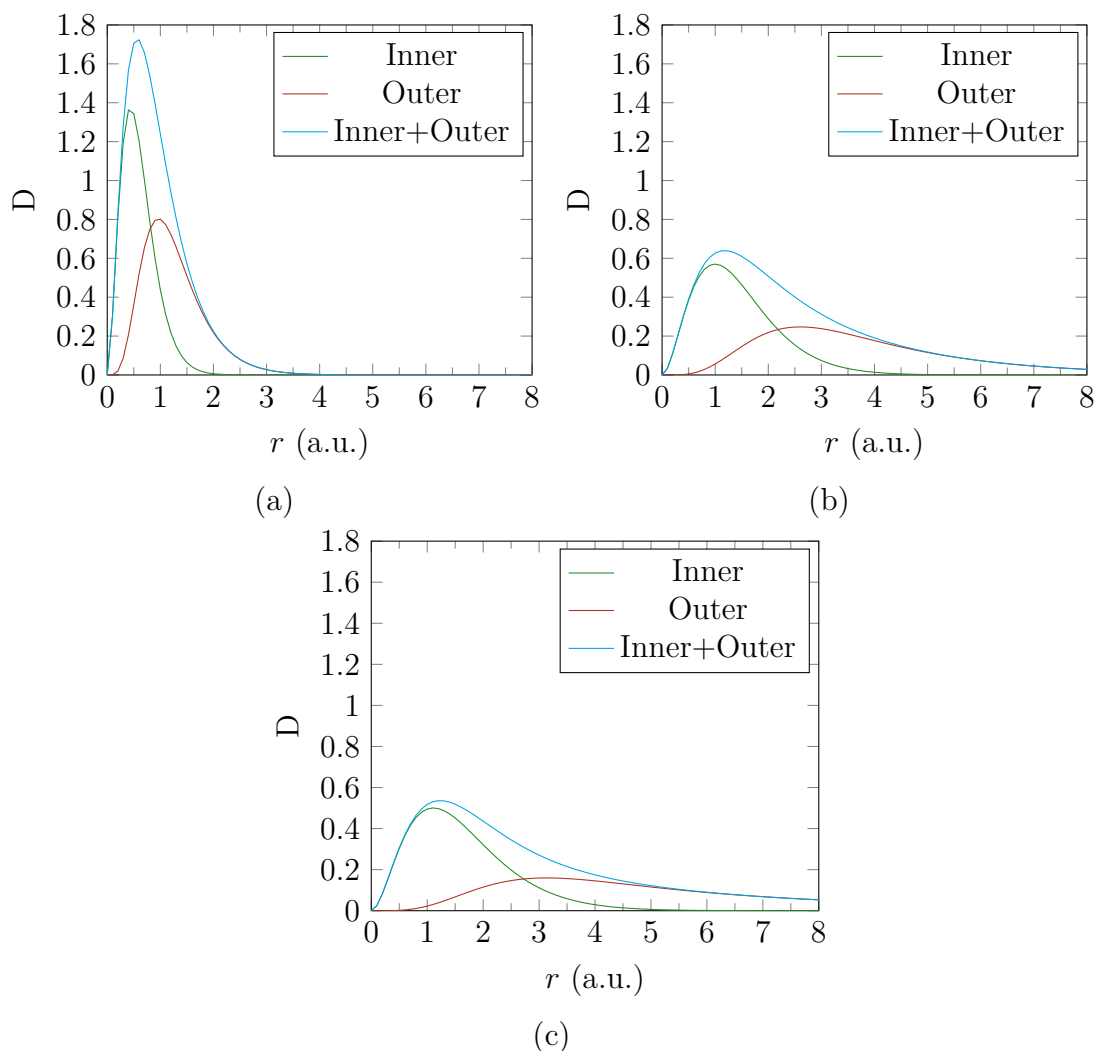
#### 6.4.4. Inner and Outer Electrons

**Particle Densities** The particle densities such as those in Figure 6.4.4 give electron-nucleus distances which represent the probability of finding either of the two electrons at distance  $r$  from the nucleus. These probabilities can be further investigated using the inner/outer electron separation technique discussed previously. At any one moment, one electron is closer to the nucleus than the other, except when  $r_1 = r_2$ . The closer electron is able to “shield” the other from the nucleus. Of course the inner electron may not be shielding the other electron from the nucleus at all times even if the electron is closer to the nucleus. For instance if the electrons are on opposite sides of the nucleus, the picture is more complex as particles are just as much waves, and behave in a probabilistic fashion. We are therefore talking about the probability of finding the particles on opposite sides of the nucleus. It should also be pointed out that “shielding” is something of a misnomer as nothing in the non-relativistic time independent Schrödinger equation stop the coulomb force from being experienced by the other particles even if they are being directly shielded. The integral for inner and outer electrons employed here as shown in Eq. (6.2.28) ranges over all possible  $r_3$  distances and that means we consider complete spheres of probability. Thus we are considering all cases where electrons are on identical and opposite sides of the nucleus thus investigating the combined effect of all positions an inner electron imposes upon an outer electron. Here the particle densities, plotted for the inner and outer electrons, measure the probability of the electron being at a certain distance. This is multiplied by its spherical shell to get the true probability density. This quantity is labelled  $D$  and is calculated as.

$$\langle D(r) \rangle = \int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} 4\pi r^2 \delta(r_i - r) dr_3 dr_2 dr_1 \quad (6.4.5)$$

Here  $r_i$  is the desired coordinate, for example  $r_1$ , or  $r_{in}$ . Figure 6.4.5 (a), (b) and (c) show the inner and outer probability densities for He,  $H^-$  and  $Z_{cr}$  respectively. In the case of He, the difference between the inner and outer electron probability density is smallest. On comparing He to  $H^-$  and  $Z_{cr}$ , it is evident that there is excess nuclear charge even after shielding. This causes the two electrons to be pulled closer together despite their repulsion. As the charge of  $Z_3$  increases, the inner and outer

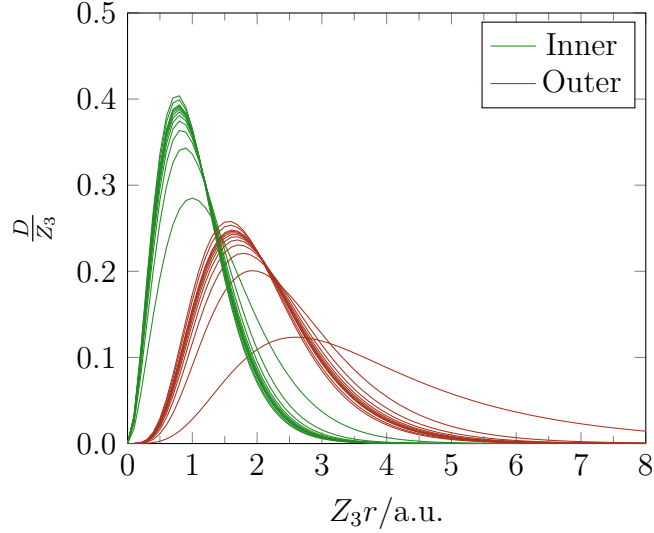
## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge



**Figure 6.4.5.:** Inner and outer radial distributions ( $D$ ) of (a) He, (b)  $H^-$ , (c)  $Z_{cr}$ .

electrons are pulled closer together. This is shown in Figure 6.4.6 where it can be seen that as  $Z_3$  increases the inner and outer electron probability densities have an increasing overlap and are pulled closer together. This figure also shows that the amount of overlap between the inner and outer electrons converges to a maximum. This theoretically correlates to a maximum amount of electron-electron interaction, whereby even after increasing the charge of the nucleus the degree of electron overlap does not increase significantly and no longer significantly increases interaction.

The  $H^-$  and  $Z_{cr}$  in Figure 6.4.5 exhibit much of the same behaviour where the outer electron is distant from the inner. This is not surprising as it is known that  $H^-$  only has a single bound state and has a difference in charge of only  $\approx 0.089$  from the critical nuclear charge, indicating that  $H^-$  is only just stable. In Chapter 7 it will be shown that even high precision HF incorrectly calculates  $H^-$  as unstable as



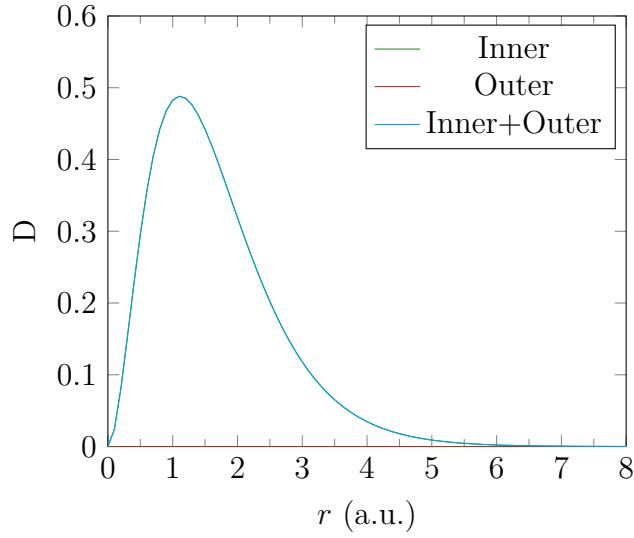
**Figure 6.4.6.:** Inner and outer radial distributions, these have been normalised in the same manner used by Thakkar *et al.*<sup>[55]</sup> such that they can be viewed on a single graph. The systems included, moving from left to right in peak heights, are  $\text{Fm}^{98+}$ ,  $\text{Ca}^{18+}$ ,  $\text{Ne}^{8+}$ ,  $\text{F}^{7+}$ ,  $\text{O}^{6+}$ ,  $\text{N}^{5+}$ ,  $\text{C}^{4+}$ ,  $\text{B}^{3+}$ ,  $\text{Be}^{2+}$ ,  $\text{Li}^{+}$ ,  $\text{He}$  and  $\text{H}^{-}$

it overestimates the important electron correlation.

At charge  $Z_3 = 0.9$  the outer electron effectively becomes unbound. This can be seen in Figure 6.4.7 where the outer electron probability is effectively zero. In this figure the inner electron radial distribution behaves exactly like the two-body radial distribution. To show this, the probability density of the two-body radial distribution is plotted on the same graph as the inner electron density plot in Figure 6.4.8. The inner electron probability densities of  $Z_{\text{cr}}$  and  $\text{H}^{-}$  also resemble the two-body probability densities. However, in these cases, the inner electron is forced closer to the nucleus compared to the two-body wavefunction. This is because the outer electron is constraining the inner electron to be closer to the nucleus to better avoid the outer electron. Hylleraas referred to such an effect as anti-shielding.<sup>[63,64]</sup>

**Radial Distances** To the authors knowledge we are the first to explicitly calculate the properties of inner and outer electrons using a wavefunction and Hamiltonian that explicitly includes the  $r_3$  term. The author does note that Koga *et al.* have calculated the inner and outer electron properties for a multi configurational HF wavefunction of atomic systems from  $\text{H}^{-}$  through to Kr. We however explicitly incorporate the electron-electron interaction through  $r_3$ . We perform the inner and outer integrals for systems with finite and infinite mass and for interesting systems

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge



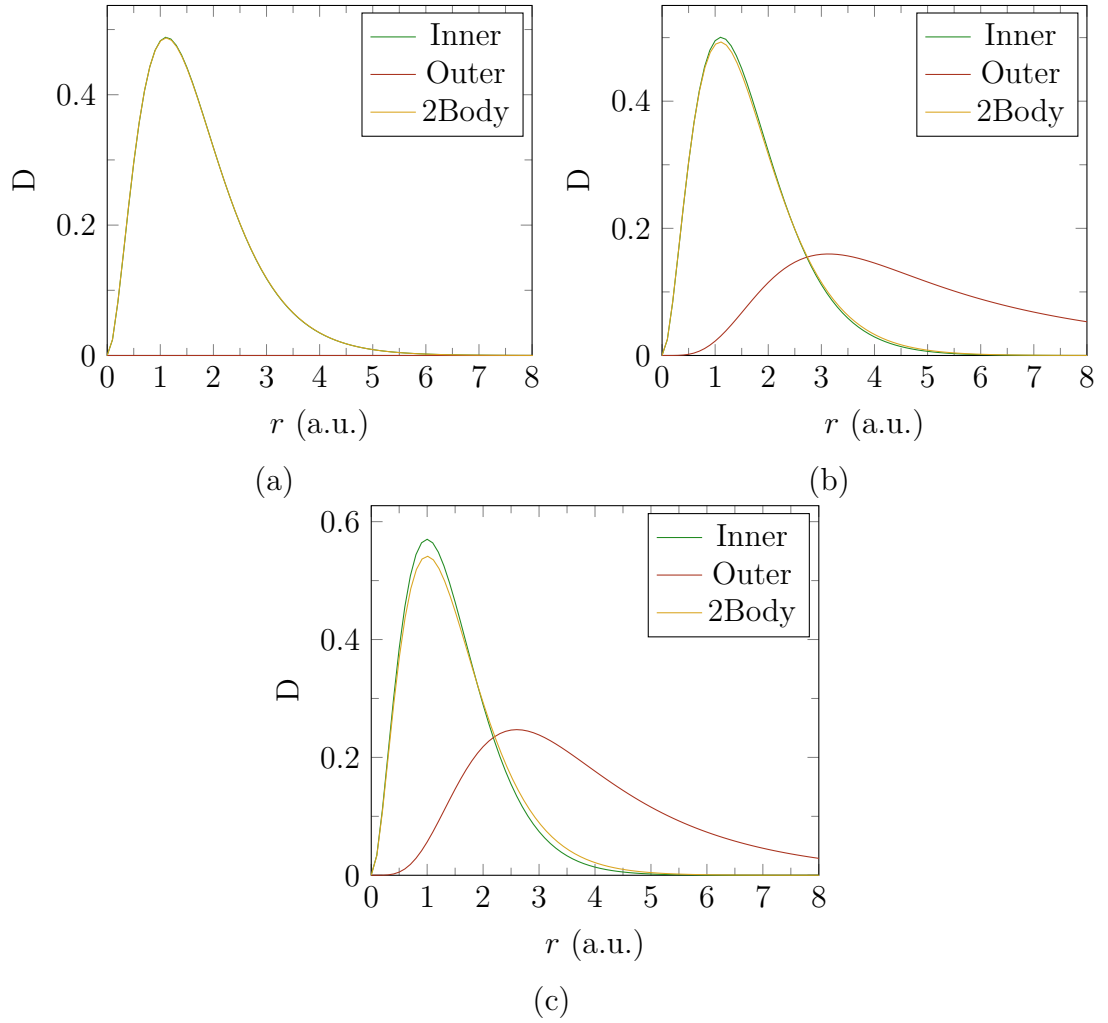
**Figure 6.4.7.:** Inner and outer radial distributions ( $D$ ) of an atom with  $m_3 = \infty$  and  $Z_3 = 0.9$ . Here Inner and (Inner+Outer) completely overlap as the outer electron density is zero (in the distribution of outer is of the scale  $1 \times 10^{-15}$  which we consider to be zero to the precision of the machine).

below  $Z_3 = 1$ , where significant differences between outer and inner electrons are observed.

The data presented in Table 6.4.4 gives the radial distances in other words the average distances the electrons are from the nucleus. It can be seen that for  $Z_{cr}$  the outer electron is still relatively well localised. However for  $Z_3 = 0.9$  the outer electron effectively resides at infinity relative to the inner electron.



6. The Stability of  $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge



**Figure 6.4.8.:** Inner and outer radial distributions ( $D$ ) of atoms with  $m_3 = \infty$  and charges of  $Z_3 = 0.9$  (a),  $Z_{cr}$  (b) and 1 (c), with the two-body (hydrogenic) radial distribution for comparison. In (a) Inner and two-body densities completely overlap.

Element	$\langle r_{inner} \rangle (a_0)$	$\langle r_{outer} \rangle (a_0)$
0.9	1.669 148 962	184.169 352 3
$Z_{cr}$	1.621 049 048	6.672 249 851
$H^-$	1.427 714 520	3.992 642 027
He	0.602 359 419	1.256 585 171
$Li^+$	0.380 512 966	0.765 035 334
$Be^{2+}$	0.278 005 362	0.550 561 294
$B^{3+}$	0.218 990 285	0.430 121 204
$C^{4+}$	0.180 638 706	0.352 949 031
$N^{5+}$	0.153 716 532	0.299 268 270
$O^{6+}$	0.133 777 667	0.259 765 875
$F^{7+}$	0.118 417 171	0.229 478 402
$Ne^{8+}$	0.106 220 610	0.205 517 571

**Table 6.4.4.:** The average distance the inner and outer electrons are from the nucleus, for a variety of atomic systems all with infinite mass.

## 6.5. Summary & Conclusions

A variational method for the direct calculation of the stability threshold to electron detachment has been introduced and proven to be very effective compared to standard methods. This method calculates the critical charge with a single computation and, for an exact wavefunction, gives an exact critical charge. This was done using our series solution method and therefore integration was avoided resulting in an efficient calculation even with a large number of terms in the wavefunction.

The minimum charge required for binding two electrons has been calculated both with (i) infinite nuclear mass and (ii) explicit consideration of nuclear motion for a range of finite nuclear masses. Taking account of the relative motion of all the particles in the system, and the non-adiabaticity and correlation effects associated with this motion,  $Z_{\text{cr}}$  changes by at most  $4 \times 10^{-5}$  for the heliogenic mass systems, but these effects become more significant,  $1 \times 10^{-2}$ , when the masses are reduced to those of “exotic” particles such as positrons and muons. Furthermore, very quickly the nuclear charge becomes insufficient to hold a bound state. This amount of charge required to reach the critical charge was found to follow the trend of fractional excess binding energy ( $g$ ) as shown in Figure 6.4.2. This is the same  $g$  that was discussed in Section 5.2.3 on page 140.

The  ${}^{\infty}Z_{\text{cr}}$  and  ${}^{\text{finite}}Z_{\text{cr}}$  for heliogenic nuclear masses agree to 4s.f. and take the value of  $0.9110e$ , which is very close to the nuclear charge of 1 in the real system,  $\text{H}^-$ . Particularly, interesting is that the energy remains smooth and continuous as  $Z_3 \rightarrow 0$ , but the expectation values of the inter-particle coordinates in the system contain a jump to large distance at just below  $Z_{\text{cr}}$ . These results indicate a transition of the system from a bound state to a shape resonance, as the nuclear charge goes through the critical point, before electron detachment. This has been published in Physical Review A.<sup>[176]</sup>

A method was conceived for the separation of the inner and outer electrons that could then be applied to various operators such as probability densities and electron-nucleus distances. This method facilitated the observation of electron detachment, that is otherwise not possible using conventional operators and the symmetric wavefunction used here as they cannot separate the two electrons. This is because they are under fermionic symmetry and therefore have identical  $\langle X(r_1) \rangle$  and  $\langle X(r_2) \rangle$

## 6. The Stability of $^1S$ -States for Coulomb 3-Body Atoms as a Function of Charge

values (where  $X(r_i)$  is an operator depending only on the  $r_i$  coordinate). This inner outer separation method has not been applied before to wavefunctions with explicit electron-electron interaction ( $r_3$ ) and also not for systems with all particles in motion.

The inner and outer electron densities were calculated for a variety of systems. As the charge was increased, the difference between the inner and outer electrons became less pronounced as the two electrons began to overlap to a greater degree. As the charge was decreased from  $Z_3 = 2$  (He) the outer electron probability began to broaden and became less associated with the nucleus, whilst the inner electron probability began to represent the two-body solution. When the charge was reduced to below this discontinuous jump, the outer electron density became flat i.e. zero probability of finding the electron anywhere, and the inner electron completely modelled the two-body hydrogenic wavefunction's probability density. This work has been submitted for publication.

Work performed in this chapter has been published in Physical Review A.<sup>[176]</sup>

# 7. Electron Correlation in Two Electron Atoms

## 7.1. Introduction

Three body systems constitute the smallest systems in which electron correlation can be observed. Electron correlation is the instantaneous interaction between two electrons. Three-body atoms such as  $\text{H}^-$ , He and their isoelectronic series have two interacting electrons in the presence of a nucleus, and represent the smallest model for such complex interactions. Electrons, being like-charged particles, repel each other. However, due to the attractive force of the nucleus, they are kept close together. The electron correlation is the complex interaction caused by these repulsive forces. These interactions are not easily calculated and many approximations exist that reduce this electron correlation to a more manageable form.<sup>[177–181]</sup> It is the effects of these approximations that we are interested in. An exact solution for three-body systems is not known, but a variational approach can be employed to obtain accuracies greater than that which would be needed to investigate electron correlation in mainstream quantum chemistry.

Electron correlation ( $E_{\text{corr}}$ ) can be defined as the difference between the energy calculated using a method in which the electron correlation is included ( $E_{FC}$ ) and that calculated from a method without the electron correlation included ( $E_{HF}$ ), calculated with an infinite basis set. Here we use the Löwdin definition<sup>[182,183]</sup> where the method without electron correlation is the HF method, as it neglects certain electron correlation and correctly describes the effects of spin.<sup>[184]</sup>  $E_{FC}$  is the energy

## 7. Electron Correlation in Two Electron Atoms

of a fully correlated system within the non-relativistic domain.

$$E_{\text{corr}} = E_{FC} - E_{HF} \quad (7.1.1)$$

Electron correlation has several effects upon a system. These include (i) the coulomb hole in which opposite-spin electrons repel each other when close and cause a small reduction in the electron probability near the nucleus. (ii) The Fermi hole in which electrons having the same spin stay far apart from each other due to the Pauli principle. In this case the electrons will not occupy the same space.<sup>[54,184–186]</sup> The electron correlation affects the separation between electrons and changes the probability of one electron being a certain distance from another.

Early work on electron correlation was directed at describing the coulomb and Fermi holes. Maslen<sup>[185]</sup> calculated the shape and size of the Fermi hole in 1956. The Fermi hole was found to have an important influence on the energy of a system particularly in metals.<sup>[184]</sup> The Fermi hole is a result of the Pauli exclusion principle and is a quantum mechanical phenomenon. The coulomb hole as reported by Coulson and Nielson<sup>[184]</sup> is a classical phenomena arising from the coulomb repulsion. The coulomb hole occurs with all electrons including with opposite spin particles where the Pauli principle has no direct influence. The coulomb hole manifests itself as an area around the electron coalescence point (as in  $r_3 = 0$ ) where the probability of finding the other electron is reduced relative to the reference electron. To completely quantify the coulomb hole, electron correlation needs to be turned off. This is where theory is a very useful technique as it allows us to switch on and off electron correlation to quantify its effects. Coulson and Nielson calculated the coulomb hole for He using the Hylleraas wavefunction, and various other wavefunctions, compared to HF. This was reported as the first numerical calculation of the coulomb hole. To calculate the coulomb hole Coulson and Nielson<sup>[184]</sup> calculated the probability of finding  $r_3$  at a specific distance. Coulson and Nielson<sup>[184]</sup> did not use the Dirac delta as used, in this and other chapters, to calculate particle densities (Section 6.3 on page 183) and instead fixed  $r_3$  to a specified value and integrated  $r_1$  and  $r_2$  over the ranges where  $r_3$  has this value. This is identical to using the Dirac delta function.<sup>[85,170]</sup> The Dirac delta function may be regarded as a shorthand way of describing this manner of integration. When we discuss the probability of  $r_3$  being

## 7. Electron Correlation in Two Electron Atoms

a specific value we will use the Dirac delta notation where  $r$  is the value  $r_3$  is to be fixed to. Coulson and Nielson concluded that the coulomb hole is  $1.1 a_0$  wide; has a minimum at  $0.54 a_0$  and moves  $0.047 e$  of charge (the charge moved is the area of the hole). From the value  $0.047 e$  it is evident how much smaller the coulomb hole is than the Fermi hole, as the Fermi hole has a charge displacement (area) of  $1.0 e$ . Recently Gill *et al.*<sup>[187]</sup> discovered a second coulomb hole for the He atom<sup>[187]</sup> and electrons in a sphere.<sup>[188]</sup> Previously thought to be due to an artefact of accuracy, Gill *et al.* have shown that it remains even with accurate wavefunctions. The primary hole at small  $r$  stabilizes the system by reducing the coulombic repulsion, however it does so at the expense of increasing the kinetic energy. In other words electron correlation allows electrons to avoid each other but to do so they must move more quickly.<sup>[187,189]</sup> Gill *et al.*<sup>[187,189]</sup> found that with the second hole the reverse is true where the kinetic energy is lowered at the expense of increased coulombic repulsion.

In 1975 Colle and Salvetti<sup>[190]</sup> created an approximate electron correlation functional using one and two-electron density matrices of He using the HF method. This functional has been used to great effect in density functional theory (DFT) with the functional LYP.<sup>[191]</sup> The Colle and Salvetti functional has also been used in a few other DFT correlation functionals.<sup>[192,193]</sup>

When investigating the effects of nuclear charge on the electron correlation of a system, the isoelectronic series provides an experimentally observable set of systems that can be investigated. The isoelectronic series of He is  $H^-$ , He,  $Li^+$ ,  $Be^{2+}$ ,  $B^{3+}$ ,  $C^{4+}$  and so on. Infinite mass is usually used through the literature,<sup>[194–196]</sup> and although our fully correlated three-body method is not limited to infinite mass, our HF implementation is. Therefore we use infinite nuclear mass throughout this chapter.

This work lies in the non-relativistic time independent regime and in this regime it has been shown that for atoms greater than carbon, the electron correlation effects plateau.<sup>[197–199]</sup> This is because the repulsion of the electrons could not be overcome with increasing charge on the nucleus. However it should be noted that within the relativistic regime, this is not the case. Styszyński *et al.*<sup>[196]</sup> investigated the relativistic He isoelectronic series's electron correlation from  $Z = 1$  to  $Z = 26$  using Hylleraas configuration-interaction (Hy-CI). Whilst Peska and Tatawaki *et al.*<sup>[200]</sup>

## 7. Electron Correlation in Two Electron Atoms

looked into the electron correlation with a greater variety of methods, from  $Z = 1$  to  $Z = 118$ , including the 4 component, multi-configurational Dirac Fock (MCDF) method.<sup>[194,200]</sup> They have shown that with relativistic effects the repulsion of the electrons is overcome and the electron correlation energy continues to change. Peska and Tatawaki *et al.*<sup>[200]</sup> report with Hy-CI and MCDF that the relativistic electron correlation energy has a more complex structure with a minimum at  $Z = 20$  and a maximum at  $Z = 68$  before rapidly decreasing. However the relativistic regime is not as simple to explore as the boundary conditions are not as well defined.<sup>[194,200]</sup>

In Chapter 4 it was shown that accurate energies for a wide range of three-body systems not limited by the Born-Oppenheimer (BO) or clamped nucleus approximation are effectively and efficiently calculated. We were also able to calculate good quality wavefunctions as indicated by the virial condition, cusps, and other factors.

In this work we will use the distribution known as the Dirac Delta function to calculate particle densities affected by electron correlation, such as is the case with coulomb holes. This is still a complex problem to be handled even on modern computers. However our choice of wavefunction, using Laguerre polynomial functions, has a set of recursion relations that simplify the task. As noted above, there has been work to calculate a functional for electron correlation in particular work by Colle and Salvetti,<sup>[190]</sup> however these functionals are not made from accurate variational calculations with full electron correlation but by analysis of HF density matrices. The work presented here uses accurate and full calculations to acquire the electron correlation. We can also investigate the effects of charge upon the electron correlation, including partial charges in ranges previously unexplored.

This chapter aims to create an accurate HF wavefunction made from the Laguerre polynomial series because of the speed of computation when using series solution and their correct cusp representation (See Section 7.2.1), calculating the analytical 2 electron integrals and developing programs and codes to effectively and efficiently create accurate HF wavefunctions. This is then used to calculate accurate electron correlation energies, by taking the fully correlated three-body energies as our  $E_{FC}$  and using the Löwdin definition for the He isoelectronic series and for non-integer charges. The effect electron correlation has on physical properties, such as average nucleus-electron distance is investigated. This includes the effects on the probability densities,

which allow us to observe the coulomb hole, and the more recently discovered second coulomb hole.

## 7.2. Method

Calculating accurate electron correlation requires the full three-body wavefunction, as discussed in detail in Chapter 4 and an accurate reference wavefunction without correlation, in this case HF. In this section the detail of the HF methodology will be given in a concise manner. We use a generic wavefunction of the form:

$$\Psi(i) = \sum_{u=0}^n c_u \varphi_u(i) \quad (7.2.1)$$

Where  $\varphi_u(i)$  forms the basis which will be specified later,  $c_u$  are weighting coefficients and  $n$  is the number of basis functions.

### 7.2.1. The Hartree-Fock Method

The HF wavefunction<sup>[101,201–203]</sup> treats every electron as existing in its own space with the nuclei. It only *sees* other electrons in an average way. This was chosen as the reference wavefunction as it is widely used and unlike a wavefunction with no electron-electron interactions it includes exchange.<sup>[203]</sup> Note that there is no exchange for  $^1S$  states of two electron atoms. More details on the general theory of Hartree-Fock are given in Chapter 2.

**The Hartree-Fock Operators** The HF Hamiltonian that describes the two electron atom with the interaction between the electrons treated in an average way has 4 operators. The first three are the standard kinetic energy  $\hat{T}$ , potential energy  $\hat{V}$  and overlap  $\hat{S}$  operators of a two-body system.

$$\hat{T}(i) = -\frac{1}{2} \left( \frac{\partial^2}{\partial r_i^2} + \frac{2}{r_i} \frac{\partial}{\partial r_i} \right) \quad (7.2.2)$$

$$\hat{V}(i) = \frac{-Z_3}{r_i} \quad (7.2.3)$$

$$\hat{S}(i) = 1 \quad (7.2.4)$$



## 7. Electron Correlation in Two Electron Atoms

The fourth operator  $\hat{G}$  deals with the coulomb and exchange interactions and treats the electron-electron interaction in an average way. These equations are in internal triangular coordinates  $(r_1, r_2, r_3)$  and not Cartesian, and therefore have a Jacobian of the coordinate transformation when performing the integrals of  $4\pi r_i^2$ . The angular momentum is separated off and the 6 Cartesian coordinates of the two electrons, relative to the nucleus, are transformed to 3 internal coordinates in a plane. When using these coordinates some aspects must be considered.

**First** The orbitals are limited to having no orbital angular momentum. This means only the S-orbital is included in our wavefunction. However as we only consider the ground state of two electron atoms, this is perfectly valid.

**Second** We are limited to atoms. This is because, in the HF implementation, only the electrons are allowed free movement. Molecules are treated in a different fashion under the BO or fixed nuclei approximation.

The fourth operator is divided into two parts, the coulombic part  $\hat{J}$  and the exchange part  $\hat{K}$

$$\hat{G}(i) = \sum_{j=1}^N 2\hat{J}_j(i) - \hat{K}_j(i) \quad i \neq j \quad (7.2.5)$$

This  $\hat{G}$  operator gives the average interaction of electron  $i$  with every other  $N - 1$  electrons. The  $\hat{J}$  is relatively simple, however the  $\hat{K}$  operator is best described by operating upon its wavefunction.

$$\hat{J}_j(i) = \langle \Psi_j(j) | \frac{1}{r_{ij}} | \Psi_j(j) \rangle \quad (7.2.6)$$

$$\hat{K}_j(i)\Psi_i(i) = \left( \langle \Psi_j(j) | \frac{1}{r_{ij}} | \Psi_i(j) \rangle \right) \Psi_j(i) \quad (7.2.7)$$

With these operator the integral contains the  $r_3 (= r_{12})$  coordinate and the Jacobian is  $8\pi^2 r_1 r_2 r_3$ .

**Fock Operator** In HF theory, the system is treated as sets of one electron functions, “seeing” the other electrons in an average way. A Fock matrix  $\hat{F}$  is created and diagonalised to obtain the energy and wavefunction of a single electron inside a two-body system. The Fock matrix is the sum of the  $\hat{T}$ ,  $\hat{V}$ ,  $\hat{G}$  and  $-\hat{S}$  matrices. To

calculate the full electronic energy of two electron atom we use the Fock operator. The Fock operator gives the energy of a single electron plus the energy of the average electron-electron interaction with every other electron. For two electrons in the same orbital we double the Fock operator's energy. However when we double the Fock operator we also double this electron-electron interaction. Therefore the electron repulsion energy is double counted. To obtain the full electronic energy, the energy calculated by the Fock operator is doubled then the double-counted electron-electron interaction energy (as calculated by  $\hat{G}$  operator) is removed.

**The SCF Cycle** With HF methodology, the Fock operator is used to calculate the wavefunction. However, the electron-electron interactions described in the  $\hat{G}$  operator require the coefficients from the wavefunction. There is therefore a dependency issue, as the  $\hat{G}$  operator requires the wavefunction and the wavefunction requires the  $\hat{G}$  operator. This is resolved with the use of an iterative cycle known as the Self Consistent Field (SCF) cycle. A guessed wavefunction is used to calculate a guessed  $\hat{G}$  operator that in turn calculates a new guessed wavefunction that in turn calculates a new guessed  $\hat{G}$  operator. This cycle repeats until the energy of the wavefunction converges to a specified tolerance. More detail of the exact procedure is given in Section 7.3.

### Hartree-Fock Methods

In this chapter three methods are used to calculate the HF energies. The first method uses the commercial computational chemistry code known as Gaussian.<sup>[204]</sup> The other two implementations use the computer algebra program Maple<sup>[205]</sup> running code written by the author with either a Gaussian wavefunction or a Laguerre polynomial wavefunction.

**Gaussian Program** The Gaussian program<sup>[204]</sup> was used to calculate HF energies in addition to the Maple implementation. This is commercial software specifically designed for calculating the properties and behaviours of atoms, molecules, their chemical processes such as reaction pathways and a large variety of other aspects. For our purposes Gaussian was used as the benchmark for the Maple implementation. Gaussian is able to calculate complex systems and interactions, however we limit

ourselves to two electron atoms with the clamped nucleus approximation. The largest Gaussian basis set available was used to acquire the most accurate results. In some cases this meant using basis sets from the basis set exchange.<sup>[206,207]</sup>

**Maple Program** With the computer algebra package Maple, two wavefunctions were implemented, namely a wavefunction made from Gaussian functions and one made from Laguerre polynomial functions. The Gaussian wavefunction was used to test the validity of the implementation prior to using the Laguerre polynomial method. This validity was confirmed by comparing the results between the Gaussian wavefunction and the Gaussian program. As the Gaussian program uses Gaussian functions, and since the wavefunction's coefficients and NLPs are available pre-optimised, we used the same coefficients and NLPs used in the Gaussian program to ensure that the results would be comparable for testing of the HF method implementation. The Laguerre polynomial wavefunction would, after testing of the Gaussian wavefunction, be used for the HF results reported here.

**Comparisons** The Gaussian program and wavefunction will both have a limitation in their accuracy as they both use pre-calculated coefficients and non-linear variational parameters ( $\alpha$ ). We could have continued to use these Gaussian-based methods and improve them with more Gaussian functions and optimised non-linear variational parameters. However, because these Gaussian-function based wavefunctions have problems such as in calculation of their cusps, we chose to use Laguerre polynomials as our basis function of choice.<sup>[208]</sup> Unlike Gaussian functions, Laguerre polynomials can better represent the cusp behaviour and so better represent the true wavefunction that has a cusp upon coalescence of particles.<sup>[105,208]</sup> For most purposes the use of Gaussian functions in the Gaussian program improves speed and still calculates energies accurately. However, in this work we desire more accurate HF wavefunctions and energies at the HF limit and also optimised wavefunctions for non-integer charges. Therefore another wavefunction was needed. A Laguerre polynomial wavefunction was chosen because Laguerre polynomials have recursion relations that can eliminate the need for some integrals; they are also the functions used in the full three-body wavefunction and so meant that minimal changes needed to be made to existing code.

## 7.3. Implementation

As described in Section 7.2.1 there are three methods for calculating a HF energy used in this work, namely the Gaussian program, and two implementations within Maple using a Gaussian wavefunction and a Laguerre polynomial wavefunction.

### 7.3.1. Gaussian Program

The Gaussian Program used version 09 rev D.01.<sup>[204]</sup> This revision allows for arbitrary nuclear charge and so could be used to calculate electron correlation energies for systems of non-integer charge. Details of this type of arbitrary charge calculation can be found in Appendix D using the **Znuc** command. The following is a simple Gaussian script used for running a calculation (will also be referred to as a “com file”) for calculating an atom.

```

The Gaussian com file for H-
1 %chk=H-.chk
2 #p hf/aug-cc-pV6Z scf=VeryTightLinEq Integral=(Acc2E=8)
3
4 IsoElectronic -1
5
6 -1 1
7 H          0.00000000  0.00000000  0.00000000
8
9

```

### 7.3.2. Maple Program

A Maple program for calculating the HF energy for an infinite mass atom with arbitrary nuclear charge was created. The Gaussian wavefunction has the following

## 7. Electron Correlation in Two Electron Atoms

generic form and it consists of normalised Gaussian functions.

$$\psi(i) = \sum_k C_k \phi_k(i) \quad (7.3.1)$$

$$\phi_k(i) = \sum_l \text{Gaussian}(i, c_{kl}, \alpha_{kl}) \quad (7.3.2)$$

$$\text{Gaussian}(i, c, \alpha) = c \left( \frac{2\alpha}{\pi} \right)^{\frac{3}{4}} e^{-\alpha r_i^2} \quad (7.3.3)$$

The following is the aug-cc-pV6Z wavefunction for He<sup>[209]</sup> from the basis set exchange.<sup>[206,207]</sup>

$$\begin{aligned} \phi_1(i) = & \text{Gaussian}(i, 0.0000006, 4785) + \text{Gaussian}(i, 0.0000047, 717) \\ & + \text{Gaussian}(i, 0.0000244, 163.2) + \text{Gaussian}(i, 0.0001012, 46.26) \\ & + \text{Gaussian}(i, 0.0003486, 15.1) + \text{Gaussian}(i, 0.0009841, 5.437) \\ & + \text{Gaussian}(i, 0.0021631, 2.088) + \text{Gaussian}(i, 0.0034905, 0.8297) \\ & + \text{Gaussian}(i, 0.003466, 0.3366) + \text{Gaussian}(i, 0.0011519, 0.1369) \end{aligned} \quad (7.3.4)$$

$$\phi_2(i) = \text{Gaussian}(i, 1, 5.437) \quad (7.3.5)$$

$$\phi_3(i) = \text{Gaussian}(i, 1, 2.088) \quad (7.3.6)$$

$$\phi_4(i) = \text{Gaussian}(i, 1, 0.8297) \quad (7.3.7)$$

$$\phi_5(i) = \text{Gaussian}(i, 1, 0.3366) \quad (7.3.8)$$

$$\phi_6(i) = \text{Gaussian}(i, 1, 0.1369) \quad (7.3.9)$$

$$\phi_7(i) = \text{Gaussian}(i, 1, 0.04473) \quad (7.3.10)$$

Equations (7.3.4) to (7.3.10) reveals a few key aspects of these Gaussian based wavefunctions. The non-linear variational parameters  $\alpha$  are all pre-optimised and do not change during the calculation. This decreases the computational expense but does limits us to wavefunctions that have been pre-calculated without creating a piece of code to optimise the non-linear variational parameters. The  $\phi_1$  ( $1^1\text{S}$ ) orbital has multiple Gaussian functions describing this function in contrast to other higher order  $\phi_i$  functions corresponding to  $2^1\text{S}$ ,  $3^1\text{S}$ , ..., etc. This wavefunction is not flexible it will only represent the orbitals for which it has been optimised.

## 7. Electron Correlation in Two Electron Atoms

**General Steps** The Maple program to perform the HF calculation has the following steps.

**Step 1** Choose a charge and define the wavefunction with any associated non-linear variational parameters.

**Step 2** Calculate the one electron integrals and assign them to matrices.

**Step 3** Diagonalise the one electron integrals as a secular equation of the form  $(\hat{T} + \hat{V} - \hat{S})\psi = 0$ . This matrix does not have electron-electron interactions and therefore does not require the SCF. We know the answer analytically for this as  $E = -\frac{2Z^2}{2n} \frac{m_1 m_3}{m_1 + m_3}$ . For the ground state with an infinite mass approximation it simplifies to  $-Z^2$ . The coefficients from this become the initial guess in the following steps.

**Step 4** Use the initial guess coefficients to create the two electron integral matrix.

**Step 5** Create the Fock matrix as the sum of the one electron and two electron matrices.

**Step 6** Diagonalise the Fock matrix as a secular equation  $\hat{F} = 0$  to acquire new coefficients.

**Step 7** Calculate the two electron integral matrix with the new coefficients.

**Step 8** Repeat the previous 3 steps until the coefficients stop changing. I.e. the SCF has converged (our tolerance for converged is  $3 \times 10^{-15}$ ). The SCF optimisation is performed using Direct Inversion of Iterative Space (DIIS), by implementing this procedure the SCF cycle converged more reliably and faster.

**Step 9** Calculate the HF energy.

**Step 10** If any NLPs are being optimised change them and return to Step 2, until the NLPs are at an optimum value (again with a tolerance of  $3 \times 10^{-15}$ ). This value is found through the inbuilt Maple optimisation routine.

The Laguerre polynomial ( $L_m(x)$ ) wavefunction was implemented in a similar

## 7. Electron Correlation in Two Electron Atoms

manner. It has the following form.

$$\psi(i) = \sum_k C_k \phi_k(i) \quad (7.3.11)$$

$$\phi_k(i) = e^{\frac{-Ar_i}{2}} L_k(Ar_i) \quad (7.3.12)$$

A single non-linear parameter  $A$  is employed. In the implementation the coordinate is changed to  $u = Ar_i$  and the appropriate Jacobian and coordinate transformations are applied to the integral. The Jacobian is  $(\frac{1}{A})$ . We do not explicitly mention the parameter in the mathematics to follow, as it is implicit in this scaled coordinate. The initial guess for the parameter is set to the formal solution of the asymptote (see Section 4.2.7 on page 100), that is, the exponential is set to  $e^{-\sqrt{E}r_i}$ .

Atomic units are used throughout, unless otherwise specified. HF was completely calculated in 32 digit arbitrary precision whilst full, three-body systems were optimised in 16 digit and then recalculated in 32 digit as a single point for accurate wavefunctions and expectation values.

### The One Electron Integrals

When using a Laguerre polynomial-based wavefunction the one electron integrals of potential energy, kinetic energy and overlap can be calculated more easily by series solution. For example the series solution method takes a very short amount of time and with a 100-term wavefunctions the computation of the integrals in the entire matrix takes less than a second. By comparison the Gaussian wavefunction without series solution takes approximately  $\approx 30$  seconds with a smaller 16 Gaussians wavefunction. The Laguerre polynomial series solution method is possible because the coordinates range from zero to infinity and the orthogonality relationship in Eq. (7.3.13) is satisfied (after the recursions relations Eq. (4.2.26) on page 96 are applied).

$$\int_0^\infty e^{-x} L_p(x) L_q(x) dx = \delta_{p,q} \quad (7.3.13)$$

The method for calculating the recursion relations and performing the series solution is the same as in Section 4.2.5 on page 96. A recursion relation for the kinetic energy,

potential energy and overlap are all calculated and these have 8 terms, 5 terms and 12 terms respectively.

Using these we can calculate the energy of He as if there was no interaction between the electrons. This value is known analytically to be  $-4 E_h$ .

### Two Electron Integrals

The two electron integrals are however more of a challenge. This is because the integrals do not range from 0 to  $\infty$  (Eq. (7.3.14)), and when converted to perimeteric coordinates, the expression is not suitable for the series solution method (Equation (7.3.15)), as the orthogonality condition is not fulfilled.

$$\int_0^\infty \int_0^\infty \int_{|r_1-r_2|}^{|r_1+r_2|} e^{-r_1-r_2} L_p(r_1) L_q(r_1) \frac{1}{r_3} L_r(r_2) L_s(r_2) dr_3 dr_2 dr_1 \quad (7.3.14)$$

$$\int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2)-z_3} L_p\left(\frac{1}{2}(z_2+z_3)\right) L_q\left(\frac{1}{2}(z_2+z_3)\right) \frac{1}{\frac{1}{2}(z_1+z_2)} L_r\left(\frac{1}{2}(z_1+z_3)\right) L_s\left(\frac{1}{2}(z_1+z_3)\right) dz_3 dz_2 dz_1 \quad (7.3.15)$$

In the two electron integral matrix each element is calculated with the following formula ( $n$  is the number of basis functions)

$$G_{uv} = \sum_{p=0}^n \sum_{q=0}^n C_p C_q (2 \langle \varphi_u(1) \varphi_v(1) | \frac{1}{r_3} | \varphi_p(2) \varphi_q(2) \rangle - \langle \varphi_u(1) \varphi_p(1) | \frac{1}{r_3} | \varphi_v(2) \varphi_q(2) \rangle) \quad (7.3.16)$$

There are two abbreviations in common use for the two electron integral's expectation value  $\langle \phi_u(1) \phi_v(1) | \frac{1}{r_3} | \phi_p(2) \phi_q(2) \rangle$  the physicist's notation  $\langle up | vq \rangle$  and the chemist notation  $(uv | pq)$ . Both represent the same integrals; however in the chemist's notation the bra contains all the basis functions on coordinate one and the ket all the functions on coordinate two. Throughout this report we use the chemist's notation.

**Integral Symmetry** The minimum number of integrals, not taking into account any symmetry, is  $n^4$  where all combinations  $pquv$  are used. Fortunately there are many repeated integrals. For example in  $(uv | pq)$  the  $u$  and  $v$  can be switched without



affecting the integrals;  $p$  and  $q$  can be switched without affecting the integral; and finally the  $uv$  and  $pq$  can be switched as a set without affecting the integral. In this implementation we therefore only calculate unique integrals. For more detail on how we chose which integrals to calculate see Appendix E.

**Computational Bottleneck** Even after calculating only the unique integrals, there are  $\frac{1}{8}n(n+1)(n^2+n+2)$  two-electron integrals to calculate. This is greater than the number of one electron integrals. It also scales much faster with number of terms, and these integrals are slower to calculate with no series solution. These integrals are by far the biggest computational bottleneck. Therefore to reduce computational time, the calculation of these integrals has been optimised.

Although the integral in Eq. (7.3.15) cannot be performed using the series solution method it can be evaluated analytically using the known analytical integral<sup>[210]</sup> in Eq. (7.3.17).

$$\int_0^\infty x^n e^{-ax} dx = \Gamma(n+1) a^{-n-1} \quad a > 0, n \geq 0 \quad (7.3.17)$$

To use this integral we must expand our Laguerre polynomials. This is done using the closed form of the Laguerre polynomials in Eq. (7.3.18).

$$L_n(x) = \sum_{n_i=0}^n (-1)^{n_i} \binom{n}{n_i} \frac{x^{n_i}}{n_i!} \quad (7.3.18)$$

Substituting Eq. (7.3.18) into Eq. (7.3.15) gives the following:

$$\begin{aligned} & \int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2)-z_3} \sum_{p_i=0}^p (-1)^{p_i} \binom{p}{p_i} \frac{\left(\frac{z_2+z_3}{2}\right)^{p_i}}{p_i!} \sum_{q_i=0}^q (-1)^{q_i} \binom{q}{q_i} \frac{\left(\frac{z_2+z_3}{2}\right)^{q_i}}{q_i!} \\ & \frac{2}{z_1+z_2} \sum_{u_i=0}^u (-1)^{u_i} \binom{u}{u_i} \frac{\left(\frac{z_1+z_3}{2}\right)^{u_i}}{u_i!} \sum_{v_i=0}^v (-1)^{v_i} \binom{v}{v_i} \frac{\left(\frac{z_1+z_3}{2}\right)^{v_i}}{v_i!} dz_3 dz_2 dz_1 \end{aligned} \quad (7.3.19)$$

## 7. Electron Correlation in Two Electron Atoms

We can then collect all the  $z_i$  variables and combine the sums to give:

$$\int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2)-z_3} \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v (-1)^{(p_i+q_i+u_i+v_i)} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i} \frac{\left(\frac{z_2+z_3}{2}\right)^{p_i+q_i}}{p_i!q_i!} \frac{\left(\frac{z_1+z_3}{2}\right)^{u_i+v_i}}{u_i!v_i!} \frac{2}{z_1+z_2} dz_3 dz_2 dz_1 \quad (7.3.20)$$

We then use the binomial expansion (Eq. (7.3.21)) on the inner most factors to get the expression in the form of the analytical integral in Eq. (7.3.17).

$$(x+y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k \quad (7.3.21)$$

$$\int_0^\infty \int_0^\infty \int_0^\infty e^{-\frac{1}{2}(z_1+z_2)-z_3} \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v (-1)^{(p_i+q_i+u_i+v_i)} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i} \frac{\sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{r_i+s_i}{b} \left(\frac{z_2}{2}\right)^{p_i+q_i-a} \left(\frac{z_3}{2}\right)^{a+b} \left(\frac{z_1}{2}\right)^{u_i+v_i-b}}{p_i!q_i!u_i!v_i!} dz_3 dz_2 dz_1 \quad (7.3.22)$$

In the next step we apply the Jacobian  $2\pi^2 r_1 r_2 r_3$  which removes the  $\frac{2}{z_1+z_2} \equiv \frac{1}{r_3}$  term and incorporates a  $2\pi^2 \left(\frac{z_1+z_3}{2}\right) \left(\frac{z_2+z_3}{2}\right) \equiv 2\pi^2 r_1 r_2$ . To deal with this the Jacobian is expanded to  $\frac{1}{2}\pi^2(z_1 z_2 + z_1 z_3 + z_2 z_3 + z_3^2)$ . However the next few steps have long expressions so we first change the form of the integral to:

$$\sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v (-1)^{(p_i+q_i+u_i+v_i)} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \binom{v}{v_i} \frac{1}{p_i!q_i!u_i!v_i!} \int_0^\infty \int_0^\infty \int_0^\infty f(p, q, u, v, p_i, q_i, u_i, v_i) dz_3 dz_2 dz_1 \quad (7.3.23)$$

## 7. Electron Correlation in Two Electron Atoms

$$f(p, q, u, v, p_i, q_i, u_i, v_i) = e^{-\frac{1}{2}(z_1+z_2)-z_3} \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{r_i+s_i}{b} \left(\frac{z_2}{2}\right)^{p_i+q_i-a} \times \left(\frac{z_3}{2}\right)^{a+b} \left(\frac{z_1}{2}\right)^{u_i+v_i-b} \quad (7.3.24)$$

We then focus on just dealing with the  $f(p, q, u, v, p_i, q_i, u_i, v_i)$  component as it is the only part with the variables from now on the rest are removed as constants. To continue we multiple this through by the Jacobian.

$$e^{-\frac{1}{2}(z_1+z_2)-z_3} \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \binom{p_i+q_i}{a} \binom{r_i+s_i}{b} \left(\left(\frac{z_2}{2}\right)^{p_i+q_i-a} \left(\frac{z_3}{2}\right)^{a+b}\right) \left(\frac{z_1}{2}\right)^{u_i+v_i-b} \times \frac{1}{2} \pi^2 (z_1 z_2 + z_1 z_3 + z_2 z_3 + z_3^2) \quad (7.3.25)$$

The expression is expanded to give:

$$e^{-\frac{1}{2}(z_1+z_2)-z_3} \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \frac{1}{2} \pi^2 \binom{p_i+q_i}{a} \binom{r_i+s_i}{b} \left(\frac{z_2}{2}\right)^{p_i+q_i-a+1} \left(\frac{z_3}{2}\right)^{a+b} \left(\frac{z_1}{2}\right)^{u_i+v_i-b+1} + \left(\frac{z_2}{2}\right)^{p_i+q_i-a} \left(\frac{z_3}{2}\right)^{a+b+1} \left(\frac{z_1}{2}\right)^{u_i+v_i-b+1} + \left(\frac{z_2}{2}\right)^{p_i+q_i-a+1} \left(\frac{z_3}{2}\right)^{a+b+1} \left(\frac{z_1}{2}\right)^{u_i+v_i-b} + \left(\frac{z_2}{2}\right)^{p_i+q_i-a} \left(\frac{z_3}{2}\right)^{a+b+2} \left(\frac{z_1}{2}\right)^{u_i+v_i-b} \quad (7.3.26)$$

At this stage we use the analytical integral in Eq. (7.3.17) on Eq. (7.3.26). Only  $f(p, q, u, v, p_i, q_i, u_i, v_i)$  need be integrated as the rest of the expression does not

## 7. Electron Correlation in Two Electron Atoms

contain any coordinates and so can be taken out as a constant.

$$\begin{aligned}
& \int_0^\infty \int_0^\infty \int_0^\infty f(p, q, u, v, p_i, q_i, u_i, v_i) dz_3 dz_2 dz_1 = \frac{1}{2} \pi^2 \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \left( \binom{p_i+q_i}{a} \right. \\
& \times \binom{r_i+s_i}{b} \left( \frac{1}{2} \right)^{p_i+q_i-a+1} \Gamma(p_i+q_i-a+2) \frac{1}{2}^{-(p_i+q_i-a+2)} \left( \frac{1}{2} \right)^{a+b} \Gamma(a+b+1) \\
& \times 1^{-(a+b+1)} \left( \frac{1}{2} \right)^{u_i+v_i-b+1} \Gamma(u_i+v_i-b+2) \frac{1}{2}^{-(u_i+v_i-b+2)} + \left( \frac{1}{2} \right)^{p_i+q_i-a} \\
& \times \Gamma(p_i+q_i-a+1) \frac{1}{2}^{-(p_i+q_i-a+1)} \left( \frac{1}{2} \right)^{a+b+1} \Gamma(a+b+2) 1^{-(a+b+2)} \left( \frac{1}{2} \right)^{u_i+v_i-b+1} \\
& \times \Gamma(u_i+v_i-b+2) \frac{1}{2}^{-(u_i+v_i-b+2)} + \left( \frac{1}{2} \right)^{p_i+q_i-a+1} \Gamma(p_i+q_i-a+2) \frac{1}{2}^{-(p_i+q_i-a+2)} \\
& \times \left( \frac{1}{2} \right)^{a+b+1} \Gamma(a+b+2) 1^{-(a+b+2)} \left( \frac{1}{2} \right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \frac{1}{2}^{-(u_i+v_i-b+1)} \\
& + \left( \frac{1}{2} \right)^{p_i+q_i-a} \Gamma(p_i+q_i-a+1) \frac{1}{2}^{-(p_i+q_i-a+1)} \left( \frac{1}{2} \right)^{a+b+2} \Gamma(a+b+3) 1^{-(a+b+3)} \\
& \times \left( \frac{1}{2} \right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \frac{1}{2}^{-(u_i+v_i-b+1)} \Bigg) \quad (7.3.27)
\end{aligned}$$

To get the complete analytical integral we insert Eq. (7.3.27) into Eq. (7.3.23):

$$\begin{aligned}
& \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v (-1)^{(p_i+q_i+u_i+v_i)} \binom{p}{p_i} \binom{q}{q_i} \binom{u}{u_i} \frac{1}{p_i! q_i! r_i! s_i!} \binom{v}{v_i} \frac{1}{2} \pi^2 \\
& \times \sum_{a=0}^{p_i+q_i} \sum_{b=0}^{u_i+v_i} \left( \binom{p_i+q_i}{a} \binom{r_i+s_i}{b} \left( \frac{1}{2} \right)^{p_i+q_i-a+1} \Gamma(p_i+q_i-a+2) \frac{1}{2}^{-(p_i+q_i-a+2)} \right. \\
& \times \left( \frac{1}{2} \right)^{a+b} \Gamma(a+b+1) 1^{-(a+b+1)} \left( \frac{1}{2} \right)^{u_i+v_i-b+1} \Gamma(u_i+v_i-b+2) + \left( \frac{1}{2} \right)^{p_i+q_i-a} \\
& \times \Gamma(p_i+q_i-a+1) \frac{1}{2}^{-(p_i+q_i-a+1)} \left( \frac{1}{2} \right)^{a+b+1} \Gamma(a+b+2) 1^{-(a+b+2)} \left( \frac{1}{2} \right)^{u_i+v_i-b+1} \\
& \times \Gamma(u_i+v_i-b+2) \frac{1}{2}^{-(u_i+v_i-b+2)} + \left( \frac{1}{2} \right)^{p_i+q_i-a+1} \Gamma(p_i+q_i-a+2) \frac{1}{2}^{-(p_i+q_i-a+2)} \\
& \times \left( \frac{1}{2} \right)^{a+b+1} \Gamma(a+b+2) 1^{-(a+b+2)} \left( \frac{1}{2} \right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \frac{1}{2}^{-(u_i+v_i-b+1)} \\
& + \left( \frac{1}{2} \right)^{p_i+q_i-a} \Gamma(p_i+q_i-a+1) \frac{1}{2}^{-(p_i+q_i-a+1)} \left( \frac{1}{2} \right)^{a+b+2} \Gamma(a+b+3) 1^{-(a+b+3)} \\
& \times \left( \frac{1}{2} \right)^{u_i+v_i-b} \Gamma(u_i+v_i-b+1) \frac{1}{2}^{-(u_i+v_i-b+1)} \Bigg) \quad (7.3.28)
\end{aligned}$$

This finally gives us the complete 2 electron integral (Eq. (7.3.14)) in analytical form which can be used to calculate Eq. (7.3.16). The results of this integral have been compared to the result Maple gives with its built in integration routines, and they agree exactly. It is however not a simple integral, it contains many factorials and summations. As a result it scales poorly with size of the wavefunction. In implementing this in Maple we see a calculation time improvement of approximately 10 seconds (12.5 %). To improve this speed further a C++ program was developed to calculate these integrals in a generic form and save them to disk. This would mean they only needed to be calculated once and could be reused, allowing us to use much larger wavefunctions as the C++ program can be made much more efficient.

**2 Electron Integrals in C++** The C++ program for calculating the two electron integrals had a lot of effort devoted to optimisation. The problem however is that the integral uses factorials, and these factorials become very large very quickly. The largest factorial is  $(4n + 4)!$  where  $n$  is the number of terms in the wavefunction. For a 20-term wavefunction this is  $84!$  which contains 127 digits. This is beyond the accuracy for built-in data types such as “double”. Therefore, we use arbitrary precision, floating point and integer arithmetic using the C++ libraries `gmp`<sup>[211]</sup> and `mpfr`<sup>[36]</sup> (details of these libraries are in Section 3.2.1). With big number arithmetic it is important to perform as few operations as possible as all operations are slow.

Consider the integral  $(12\ 12|12\ 12)$ . Within Maple this integral takes approximately 80 seconds. Using the mathematics in the previous section this integral originally took 70 seconds in C++. This was far from ideal and so a series of optimisations was performed with the target of reducing the computational time:

- The code was profiled for CPU time using XCode instruments and it was discovered that a lot of CPU time was devoted to the creation and destruction of the arbitrary precision floating point numbers. This was resolved by reusing the variables, rather than recreating them, in the main for loops that represent the sums.
- The next optimisation reduced the binomial and gamma functions to factorials and made the mathematics simpler with fewer operations (The new expression is given in Eq. (7.3.29)).

## 7. Electron Correlation in Two Electron Atoms

- We also precomputed all of the factorials and saved them in an array for lookup in the calculations.

The two electron integral now has the form:

$$\begin{aligned}
& \sum_{p_i}^p \sum_{q_i}^q \sum_{u_i}^u \sum_{v_i}^v \sum_{a_i}^{p_i+q_i} \sum_{b_i}^{u_i+v_i} 2^{a_i+b_i-p_i-q_i-u_i-v_i+1} (-1)^{p_i+q_i+u_i+v_i+1} \pi^2 (a_i^2 - 2a_i b_i \\
& + b_i^2 - p^2 - 2p_i q_i - 2p_i u_i - 2p_i v_i - q_i^2 - 2q_i u_i - 2q_i v_i - u^2 - 2u_i v_i - v^2 + a_i \\
& + b_i - 7p_i - 7q_i - 7u_i - 7v_i - 10) (p_i + q_i + u_i + v_i - a_i - b_i)! (p_i + q_i)! (u_i + v_i) \\
& \times \frac{!p!q!u!v!}{u_i!^2 v_i!^2 u_i!^2 v_i!^2 (p_i + q_i - a_i)! (q_i + u_i - b_i)! (p - p_i)! (q - q_i)! (u - u_i)! (v - v_i)!} \\
& \tag{7.3.29}
\end{aligned}$$

In this form there are fewer operations to perform. It is worth noting that if faster computation of the two-electron integral is required, this form can be further factorised to reduce the number of operations.

The CPU time was now approximately 15 seconds. As a final step the code was made to run in parallel with multiple cores, reducing the CPU time to 8 seconds, achieving a 10-fold speed increase. This is the time required to calculate one two-electron integral. For a twenty-term wavefunction there are, in total, 22 155 two-electron integrals. The time required to calculate all these integrals therefore is still significant and justifies further optimisation.

When calculating all unique two electron integrals, for a 12 term wavefunction, we iterate over all unique combinations of the complex integrals in  $(pq|uv)$  for  $p, q, u, v$  less than 12. When calculating the integral  $(12\ 12|12\ 12)$  almost all of the mathematics for the lower integrals i.e.  $(6\ 6|6\ 6)$ ,  $(1\ 2|1\ 1)$  and so on are also performed. It follows then that we can drastically reduce the computational expense by performing the largest integral first and reusing the numbers it calculates by saving the intermediate steps to memory. This does however need a substantial amount of RAM. A 20 term wavefunction uses 10 GB of RAM in this method, which is near the maximum of our available resources. However the method was over 20 times faster than calculating all the integrals individually.

This concludes the description of the optimisation techniques developed to maximise the performance of the two-electron integral calculations. More optimisations

are possible that may be pursued at a later date. For example it should be possible to combine the loops calculating all combinations of  $(pq|rs)$  with the loops that represent the sums. By doing so it would be possible to get most of the benefits of the RAM-intensive mode without saving as much to the RAM. This would however require a major rewrite and careful planning.

Even after calculating all unique two electron integrals in C++, a significant amount of time is still required to load these integrals into Maple and assign the integrals to their correct positions in the  $\hat{G}$  matrix. Therefore a final optimisation was made to the Maple program to save the final  $\hat{G}$  matrix to disk. This file is still fully generic and reduces the calculation time for subsequent calculations from the first time it is performed. The largest number of terms currently calculated by C++ for the Laguerre polynomial wavefunction is 25 terms. Calculating the two-electron integrals for the 25-term wavefunction took significantly longer than for the 20-term wavefunction. It also took significantly longer to load the two-electron integrals into Maple and assign into the  $\hat{G}$  matrix. This was because the memory requirements exceeded the limit of the computer (16GB). Therefore the largest wavefunction used in this work has 20 terms. Unless otherwise stated all Laguerre polynomial wavefunctions have 20 terms.

To summarise, three methods were used to calculate HF wavefunctions and energies, (i) using the Gaussian program, (ii) using the Maple program with Gaussian wavefunctions and (iii) using the Maple program with Laguerre polynomial wavefunctions. The Laguerre polynomial wavefunction was our best choice for calculating accurate wavefunctions for a variety of reasons including correct behaviour when the particles coalesce (the cusps). The one-electron integrals are calculated efficiently with series solution and the two-electron integrals are calculated efficiently with an optimised C++ code.

### 7.4. Results & Discussion

The HF wavefunction is the reference wavefunction for calculating electron correlation effects. For this purpose the HF wavefunction needs to be as accurate as possible. Therefore we performed benchmarks to validate the methods including

checking the convergence and comparing to known systems such as  $\text{H}^-$ , He and other members of the helium isoelectronic series under the infinite mass approximation. Using the methods described in Sections 7.2 and 7.3 the Hartree-Fock energies were calculated with the single non-linear variational parameter Laguerre polynomial based wavefunction (Method A), with the Gaussian wavefunction (Method B) and the Gaussian program (Method C).

### 7.4.1. Method Comparison

In Section 7.2.1 Method B was said to be a test of the Maple program method for calculating HF energies. This test would be passed successfully if Method B could match the energies predicted by the commercial Gaussian program (Method C). When the above method was able to achieve these results, we would then use our own Laguerre polynomial-based wavefunction (Method A). Here we provide the results of this testing. In Table 7.4.1, the results of each method are presented and compared to the numerical HF results for  $\text{H}^-$  and He. The numerical HF method<sup>[212,213]</sup> is equivalent to an infinite basis set approach, and is accurate up to the limit of the integration grid and numerical precision.

The energies calculated with Method B matched that of Method C to 9s.f., the differences arising from the SCF procedure, as the Gaussian program used quadratic convergent optimisation<sup>[216]</sup> whilst the Maple implementation used DIIS. Additionally Method B produced the same eigenvalues as Method C and the same eigenvectors. Method A was then developed and found that with 20 terms in the wavefunction it matched the numerical HF results and gave significantly more accurate energies than the Gaussian HF energies. From here on we do not provide Method B results, using instead Method C when a Gaussian comparison is required as it has all the Gaussian wavefunctions preloaded and ready to use.



Species	Method	Energy
$\text{H}^-$	A	−0.487 929 734 369
	B	−0.487 909 650 178
	C	−0.487 909 650 182
	Numerical HF <sup>[214]</sup>	−0.487 929 734 72
He	A	−2.861 679 995 612
	B	−2.861 673 128 997
	C	−2.861 673 128 997
	Numerical HF <sup>[215]</sup>	−2.861 679 995 612

**Table 7.4.1.:** A comparison of the different methods for calculating HF energies. Method A, Laguerre polynomial wavefunction using Maple. Method B, Gaussian wavefunction using Maple. Method C, Gaussian wavefunction using the program Gaussian.

### 7.4.2. Convergence

Method A was capable of achieving numerical HF with 20 terms. This means the wavefunction has already achieved our best benchmark. To identify how many more digits we have calculated beyond the numerical HF results, we studied the convergence rate of the wavefunction. In Table 7.4.2 the convergence rate of the energy is provided for He, compared to numerical HF results.

The results of Table 7.4.2 shows that the energy converges towards the exact numerical HF value, to all significant figures available, with only 16 terms in the wavefunction. It can be seen that the rate of convergence of the Hartree-Fock energies is much greater than that of the full, three-body systems with the explicit  $r_3$  interaction. In the full three-body calculations, energy convergence to 8 digits requires  $\approx 715$  terms in the wavefunction, whilst the Hartree-Fock wavefunction requires only  $\approx 10$  terms in the wavefunction.

From these data and that in Table 7.4.1 we can therefore confidently say that we can calculate accurate HF energies with this method.

No. of Terms in $\psi$	Number of 2e Integrals	NLP	2Body Energy	HF Energy
1	1	3.375	-3.902 343 749 999 996 763 517 840 730 0	-2.847 656 249 999 999
3	21	3.842	-3.999 999 993 527 671 251 524 220 504 9	-2.861 590 054 663 477
5	120	3.933	-3.999 999 999 999 999 999 562 829 282 4	-2.861 679 675 254 110
7	406	3.509	-3.999 999 999 999 988 428 099 458 088 4	-2.861 679 763 443 967
10	1540	4.997	-3.999 999 999 999 932 956 115 553 4	-2.861 679 993 922 506
12	3081	5.218	-3.999 999 999 999 999 036 851 048 6	-2.861 679 995 536 817
14	5565	5.384	-3.999 999 999 999 999 991 496 449 6	-2.861 679 995 609 217
16	9316	5.506	-3.999 999 999 999 999 999 948 485 7	-2.861 679 995 612 136
18	14 706	5.581	-3.999 999 999 999 999 999 999 823 9	-2.861 679 995 612 237
20	22 155	5.582	-3.999 999 999 999 999 999 999 999 8	-2.861 679 995 612 239
Exact/Lit	N/A	N/A	-4	-2.861 679 995 612

**Table 7.4.2.:** The effects of wavefunction size on the accuracy of the wavefunction. Detailing the rate of convergence of the HF energy, Compared to literature HF.<sup>[215]</sup>

### 7.4.3. Non-linear Variational Parameters

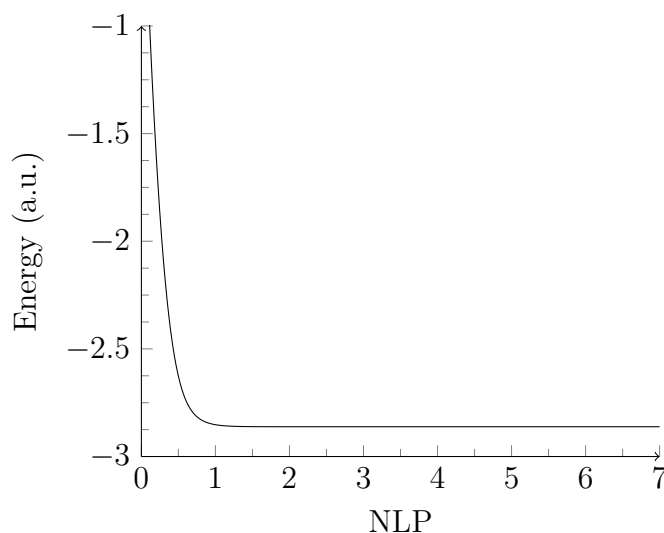
Method A has a single, non-linear variational parameter (NLP). The effects this NLP has on the energy, and the degree to which we can find the optimum values, of the NLP are important for the energy provided by the calculation. Table 7.4.3 shows the effect of the value of the NLP on the HF energy with a value of 1,  $Z_3$  (which is 2 for He) and optimised (which can be any real positive number) with 10 and 20-term He wavefunctions. When the NLP has a value of 1 the wavefunction has the exponent of  $e^{-r}$ ; when it is  $Z_3$  then the exponent becomes  $e^{-rZ_3}$  (for He it is  $e^{-2r}$ ). In this case the exponent represents the asymptotic solution for the non-interacting particle model, as discussed in Section 4.2.7 on page 100.

Table 7.4.3 shows that the value of the NLP is very important in order to achieve the accuracy of the numerical HF results as it greatly improves the energy of the wavefunction. The 10-term wavefunction gained an additional 4 digits and 20 terms gained 5 additional converged digits matching the literature.

It is therefore important that we optimise the parameter as much as is possible to achieve the best energies with the smallest wavefunction. Figure 7.4.1 shows the scan of the NLP for the 10-term wavefunction, and it shows that it is relatively flat. Flat surfaces are typically difficult to optimise to a definitive minimum. The first optimisation routine we used was the conjugate gradient routine (built into Maple). This routine failed to optimise the NLP to the bottom of the well. After trying various optimisation routines it was found that the Quadratic Interpolation routine achieved the best optimisation. To demonstrate this, Figure 7.4.2 shows how effective the routine is at reaching this minimum by zooming in at the minimum of

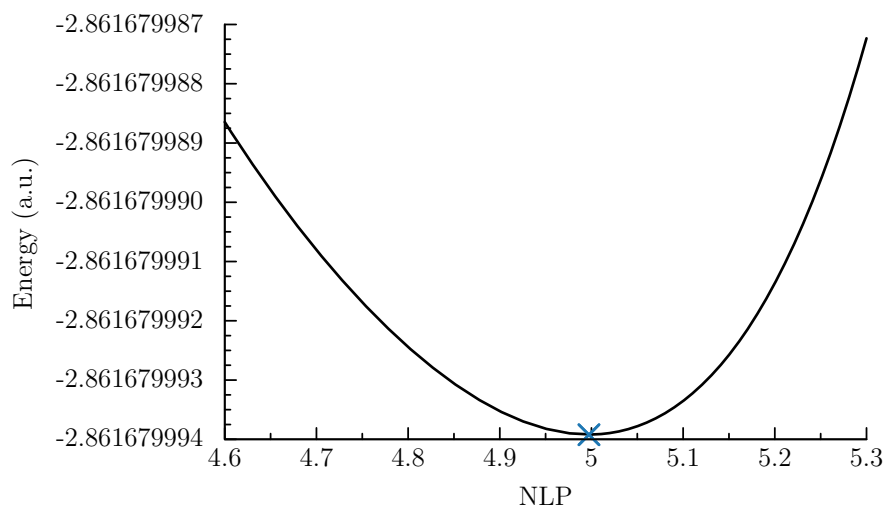
Num of terms in $\psi$	NLP	HF Energy (a.u.)
10	1	−2.852 188 594 931
	2	−2.861 664 390 672
	4.997 241 294	−2.861 679 993 922
20	1	−2.861 643 004 528
	2	−2.861 679 942 413
	5.581 838 558	−2.861 679 995 612
Numerical HF	N/A	−2.861 679 995 612

**Table 7.4.3.:** The effect of the non-linear variational parameter on the energy of a 10 and 20-term He wavefunction.



**Figure 7.4.1.:** This is the effect the NLP has on the energy of the He system with a 10-term wavefunction.

Figure 7.4.1. This method requires finite bounds on the minimum as input. The values we provided were 0.1 to  $3Z_3^2$ . Figure 7.4.2 is a small region of Figure 7.4.1 the cross represents where the results of the optimisation method. The cross in Figure 7.4.2 shows that the optimised point is near the very bottom of the energy surface. We therefore use this quadratic optimisation method in all HF NLP optimisation.



**Figure 7.4.2.:** This is a small region of the Figure 7.4.1 it shows the optimisation result as a cross

#### 7.4.4. Hartree-Fock Energies

The energies calculated using the HF wavefunction are used in Section 7.4.6 to calculate the electron correlation of the system. In this section we evaluate the accuracy of this energy by comparing to literature HF energies of the He isoelectronic series. Methods A and C are compared in Table 7.4.4 to these literature values.

From Table 7.4.4 we know that the energies from Method A are very accurate achieving the numerical Hartree-Fock values to the digits reported. Method C energies however are accurate to at most 7 s.f.. With Method C, 7 s.f. is more than enough for most needs. However, in this work we will be looking at less-studied theoretical systems with arbitrary nuclear charge. At these arbitrary charges, given the pre optimised nature of the basis sets, it will be difficult to maintain accuracy with Method C for non-integer unit charges as there are no readily available basis sets specialised for these arbitrary charges. The values calculated here with Method A agree with the numerical HF results. As Method A is very accurate, and with its non-linear variational parameter, it is flexible and will be the primary method in the following work.

## 7. Electron Correlation in Two Electron Atoms

Species	Hartree Fock Energy (au)			
	Method A	Method C	Literature HF	
H <sup>-</sup>	-0.487 929 734 369 2	-0.487 909 7 <sup>a</sup>	-0.487 929 734 372	[214]
He	-2.861 679 995 612	-2.861 673 1 <sup>a</sup>	-2.861 679 995 6	[217]
Li <sup>+</sup>	-7.236 415 201 452	-7.236 384 5 <sup>b</sup>	-7.236 415 201 431 42	[218]
Be <sup>2+</sup>	-13.611 299 430 62	-13.611 251 5 <sup>b</sup>	-13.611 299 430 604	[219]
B <sup>3+</sup>	-21.986 234 466 82	-21.986 231 1 <sup>a</sup>	-21.986 234 466 814 1	[219]
C <sup>4+</sup>	-32.361 192 875 72	-32.361 188 3 <sup>a</sup>	-32.361 192 875 710 1	[219]
N <sup>5+</sup>	-44.736 163 964 94	-44.736 157 8 <sup>a</sup>	-44.736 163 964 935 6	[219]
O <sup>6+</sup>	-59.111 142 701 91	-59.111 134 8 <sup>a</sup>	-59.111 142 701 907 8	[219]
F <sup>7+</sup>	-75.486 126 406 26	-75.486 116 5 <sup>a</sup>	-75.486 126 406 255 1	[219]
Ne <sup>8+</sup>	-93.861 113 519 23	-93.861 101 3 <sup>a</sup>	-93.861 113 519 195 7	[219]
Na <sup>9+</sup>	-114.236 103 072 5	-114.232 789 <sup>b</sup>	-114.236 103 1	[220]
Mg <sup>10+</sup>	-136.611 094 432 9	-136.607 555 <sup>b</sup>	-136.611 094 4	[220]
Al <sup>11+</sup>	-160.986 087 168 8	-160.983 784 <sup>a</sup>	-160.986 087 2	[220]
Si <sup>12+</sup>	-187.361 080 975 9	-187.358 377 <sup>a</sup>	-187.361 081 0	[220]
P <sup>13+</sup>	-215.736 075 633 6	-215.733 129 <sup>a</sup>	-215.736 075 6	[220]
S <sup>14+</sup>	-246.111 070 977 8	-246.107 892 <sup>a</sup>	-246.111 071 0	[220]
Cl <sup>15+</sup>	-278.486 066 884 2	-278.482 892 <sup>a</sup>	-278.486 066 9	[220]
Ar <sup>16+</sup>	-312.861 063 256 8	-312.857 854 <sup>a</sup>	-312.861 063 3	[220]

<sup>a</sup> Calculated using aug-cc-pV6Z

<sup>b</sup> Calculated using aug-cc-pVQZ

**Table 7.4.4.:** The Hartree-Fock energies calculated here compared to best literature available

### 7.4.5. Low Nuclear Charge Hartree-Fock

To demonstrate that the Gaussian wavefunction of Method C fails to maintain its accuracy at low nuclear charge, we calculated the energies using methods A and C at the critical nuclear charge ( $Z_{\text{cr}} = 0.9110282$ ) in Table 7.4.5 (see Chapter 6 for details on critical charge). With the full three-body wavefunction this nuclear charge of  $Z_{\text{cr}}$  gives an energy equal to the two-body energy as it lies at the lowest continuum threshold. With the HF wavefunction, according to Table 7.4.5, a system with this charge gives an energy less stable than the two-body energy ( $-0.414\,986\,E_{\text{h}}$ ) and is therefore already unstable with respect to the lowest continuum threshold, as in this corresponds to electron detachment. In Section 7.4.9 we will show how the critical nuclear charge for the HF method is higher than that in a fully correlated method. Table 7.4.5 reveals that Method C, has a more positive energy than Method C in as few as the 4<sup>th</sup> s.f.. The HF wavefunction is variational (see Section 2.6 on page 11 for details on variational principle) and therefore the most negative energy is the most stable energy and the most accurate value. As we are interested in accurate energies and accurate expectation values of the wavefunction, the 4<sup>th</sup> s.f. is not accurate enough for our needs.

**Low  $Z_3$  Convergence** The arbitrary nuclear charge energies have no literature reference by which to judge the accuracy of our code. We know that for the He isoelectronic series we have very accurate energies at the HF limit. Does this extend to charges lower than unity and if not, at what point does the accuracy drop significantly? To answer this question we performed a convergence study with charges 1, 0.9, 0.8, 0.5, 0.2 and 0.1 with wavefunctions containing 20, 16, 12, 8 and 4 terms. In Table 7.4.6, we can see that the convergence rate rapidly drops as the nuclear charge decreases. When  $Z_3 = 1$  we achieve 9 s.f. of convergence with the

Wavefunction	HF Energy	Electron Correlation
Method A	$-0.373\,904\,785$	$0.041\,081\,438\,7$
Method C	$-0.373\,709\,945$	$0.041\,276\,279\,0$

**Table 7.4.5.:** The HF energies for a charge of  $Z_{\text{cr}}$  using Method A and C. This demonstrates that Method C is not the best choice for such non-integer charges.

## 7. Electron Correlation in Two Electron Atoms

Z	Terms	HF Energy	Z	Terms	HF Energy
1	4	−0.487 494 422 733 26	0.5	4	−0.051 455 832 250 40
	8	−0.487 929 420 478 26		8	−0.053 545 582 831 15
	12	−0.487 929 728 147 82		12	−0.053 470 675 489 69
	16	−0.487 929 734 257 90		16	−0.053 440 234 665 00
	20	−0.487 929 734 369 18		20	−0.053 425 905 116 87
	Lit <sup>[214]</sup>	−0.487 929 734 372			
0.9	4	−0.360 244 491 909 96	0.2	4	0.003 086 197 285 25
	8	−0.360 886 611 825 98		8	0.003 562 665 730 44
	12	−0.360 888 381 644 19		12	0.003 679 232 222 64
	16	−0.360 888 402 143 18		16	0.003 727 592 405 19
	20	−0.360 888 403 143 57		20	0.003 752 590 335 34
0.8	4	−0.253 078 124 310 81	0.1	4	0.006 295 019 883 54
	8	−0.254 177 026 783 66		8	0.002 773 834 911 73
	12	−0.254 204 267 742 55		12	0.002 410 889 973 13
	16	−0.254 217 315 751 26		16	0.002 438 246 397 61
	20	−0.254 245 116 965 40		20	0.002 451 177 471 85

**Table 7.4.6.:** HF energies and electron correlations for low  $Z$

20 term wavefunction and match literature to 10 s.f.. From the convergence and comparison to literature we can see that the significant figures that are not changing between 16 and 20 terms indicates that we are converged to 9 s.f.. When  $Z_3$  has a charge of 0.9 this reduces to 8 s.f. and when  $Z_3 = 0.8$  convergences rapidly reduces to 4 s.f.. We therefore conclude that energies are not sufficiently reliable below  $Z_3 = 0.9$ .

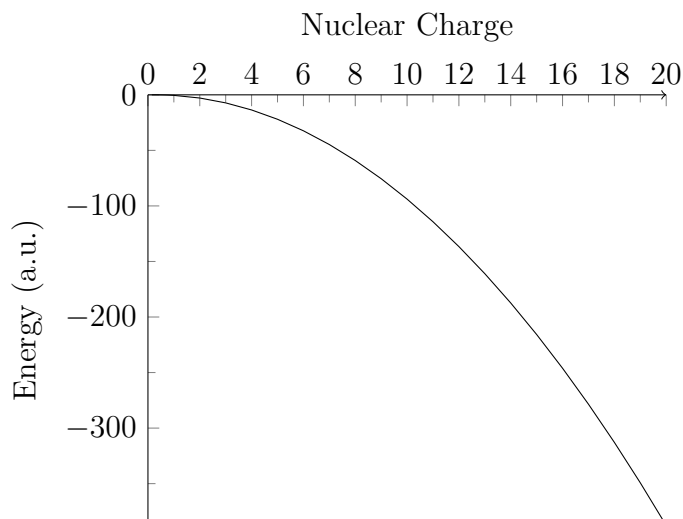
**Energies** We are now in a position to calculate HF energies for low values of  $Z_3$ . For this purpose we calculated the energies of systems with  $Z_3$  ranging from 0.99 to 0.90. The following graph shows the HF energy ( $E_{HF}$ ) as a function of  $Z_3$ . This graph includes low values of  $Z_3$  where the HF wavefunction is unstable with respect to its lowest continuum threshold.

From Figure 7.4.3 the HF energy calculated using Method A moves smoothly to zero as the charge tends towards zero. This is the correct behaviour for HF energies. These energies will be used to calculate electron correlation energies in the following sections.



Nuclear Charge $Z_3$	HF Energy
0.99	−0.474 317 036
0.98	−0.460 905 894
0.97	−0.447 696 415
0.96	−0.434 688 716
0.95	−0.421 882 929
0.94	−0.409 279 199
0.93	−0.396 877 687
0.92	−0.384 678 575
0.919	−0.383 469 804
0.918	−0.382 263 059
0.917	−0.381 058 340
0.916	−0.379 826 352
0.915	−0.378 654 983
0.914	−0.377 456 345
0.913	−0.376 259 735
0.912	−0.375 065 152
0.911	−0.373 872 597
0.91	−0.372 682 070
0.909	−0.371 462 042
0.908	−0.370 275 235
0.907	−0.369 090 452
0.906	−0.367 907 693
0.905	−0.366 726 959
0.904	−0.365 548 249
0.903	−0.364 371 564
0.902	−0.363 196 904
0.901	−0.362 024 269
0.9	−0.360 853 659

**Table 7.4.7.:** HF energies for low  $Z_3$



**Figure 7.4.3.:** Graph of HF energy ( $E_{HF}$ ) against nuclear charge ( $Z_3$ ). Showing how the HF energy changes as  $Z_3$  tends towards zero

### 7.4.6. Total Electron Correlation Energies

We next investigate  $E_{\text{corr}}$ . The Löwdin definition of electron correlation<sup>[182,183]</sup> is that electron correlation is the difference between the HF energy and the exact energy, which is taken here as the fully correlated, three-body energy with infinite nuclear mass under the non-relativistic regime. We abbreviate this fully-correlated energy as FC energy. The electron correlation at  $Z_{\text{cr}}$  can be found in Table 7.4.5. The FC energies required to calculate the electron correlation for the clamped nucleus, helium isoelectronic series were calculated with a 2856-term wavefunction. Comparison of these to the HF energies given in Section 7.4.4 gives the electron correlation. Additionally the non-integer charges were investigated around  $Z_{\text{cr}}$ .

Tables 7.4.8 and 7.4.9 gives the total electron correlation of three-body infinite nuclear mass systems with varying nuclear charge. High charge systems such as  $\text{Ca}^{18+}$  have the smallest fraction of electron correlation compared to the total energy whilst lower charge systems such as  $\text{H}^-$  has the most electron correlation as a fraction of the total energy. It is shown in Table 7.4.8 that  $\text{H}^-$  Hartree-Fock energy is greater than its lowest continuum threshold of  $-0.5 E_{\text{h}}$  and is therefore according to Hartree-Fock theory unstable (for more detail on stability as a function of charge see Chapter 6).

Species	HF Energy (a.u.)	FC Energy (a.u.)	$E_{\text{corr}}$ (a.u.)	Lit. $E_{\text{corr}}$ (a.u.)	$E_{\text{corr}}$ as % of FC Energy
H <sup>-</sup>	-4.879 297 34 $\times 10^{-1}$	-5.277 510 17 $\times 10^{-1}$	3.982 128 22 $\times 10^{-2}$	3.982 $\times 10^{-2}$ [221]	7.545 47
He	-2.861 680 00	-2.903 724 38	4.204 438 14 $\times 10^{-2}$	4.205 $\times 10^{-2}$ [221]	1.447 95
Li <sup>+</sup>	-7.236 415 20	-7.279 913 41	4.349 821 12 $\times 10^{-2}$	4.350 $\times 10^{-2}$ [221]	0.597 510
Be <sup>2+</sup>	-1.361 129 94 $\times 10^1$	-1.365 556 62 $\times 10^1$	4.426 680 78 $\times 10^{-2}$	4.430 $\times 10^{-2}$ [191]	0.324 167
B <sup>3+</sup>	-2.198 623 45 $\times 10^1$	-2.203 097 16 $\times 10^1$	4.473 711 34 $\times 10^{-2}$	4.480 $\times 10^{-2}$ [221]	0.203 065
C <sup>4+</sup>	-3.236 119 29 $\times 10^1$	-3.240 624 66 $\times 10^1$	4.505 372 62 $\times 10^{-2}$		0.139 028
N <sup>5+</sup>	-4.473 616 40 $\times 10^1$	-4.478 144 51 $\times 10^1$	4.528 118 38 $\times 10^{-2}$		0.101 116
O <sup>6+</sup>	-5.911 114 27 $\times 10^1$	-5.915 659 51 $\times 10^1$	4.545 242 08 $\times 10^{-2}$		0.076 834 1
F <sup>7+</sup>	-7.548 612 64 $\times 10^1$	-7.553 171 24 $\times 10^1$	4.558 595 77 $\times 10^{-2}$		0.060 353 4
Ne <sup>8+</sup>	-9.386 111 35 $\times 10^1$	-9.390 680 65 $\times 10^1$	4.569 299 58 $\times 10^{-2}$	4.570 $\times 10^{-2}$ [221]	0.048 657 8
Na <sup>9+</sup>	-1.142 361 03 $\times 10^2$	-1.142 818 84 $\times 10^2$	4.578 070 35 $\times 10^{-2}$		0.040 059 5
Mg <sup>10+</sup>	-1.366 110 94 $\times 10^2$	-1.366 569 48 $\times 10^2$	4.585 387 97 $\times 10^{-2}$		0.033 554 0
Al <sup>11+</sup>	-1.609 860 87 $\times 10^2$	-1.610 320 03 $\times 10^2$	4.591 585 72 $\times 10^{-2}$		0.028 513 5
Si <sup>12+</sup>	-1.873 610 81 $\times 10^2$	-1.874 070 50 $\times 10^2$	4.596 902 27 $\times 10^{-2}$		0.024 529 0
P <sup>13+</sup>	-2.157 360 76 $\times 10^2$	-2.157 820 91 $\times 10^2$	4.601 512 99 $\times 10^{-2}$		0.021 324 8
S <sup>14+</sup>	-2.461 110 71 $\times 10^2$	-2.461 571 26 $\times 10^2$	4.605 549 64 $\times 10^{-2}$		0.018 709 8
Cl <sup>15+</sup>	-2.784 860 67 $\times 10^2$	-2.785 321 58 $\times 10^2$	4.609 113 11 $\times 10^{-2}$		0.016 547 9
Ar <sup>16+</sup>	-3.128 610 63 $\times 10^2$	-3.129 071 86 $\times 10^2$	4.612 281 96 $\times 10^{-2}$		0.014 740 1

**Table 7.4.8.:** The total electron correlation for the infinite nuclear mass He isoelectronic series according to the Löwdin definition of electron correlation [182,183]

Nuclear Charge	HF Energy (a.u.)	Full 3Body Energy (a.u.)	$E_{\text{corr}}$ (a.u.)	$E_{\text{corr}}$ as % of Full 3Body Energy
0.9	$-3.608\,884\,03 \times 10^{-1}$	$-4.044\,093\,57 \times 10^{-1}$	$4.352\,095\,35 \times 10^{-2}$	10.76
0.91	$-3.726\,820\,70 \times 10^{-1}$	$-4.137\,992\,09 \times 10^{-1}$	$4.111\,713\,98 \times 10^{-2}$	9.936
$Z_{\text{cr}}\,0.911\,028$	$-3.739\,062\,41 \times 10^{-1}$	$-4.149\,862\,23 \times 10^{-1}$	$4.107\,998\,18 \times 10^{-2}$	9.899
0.92	$-3.846\,785\,75 \times 10^{-1}$	$-4.254\,852\,82 \times 10^{-1}$	$4.080\,670\,62 \times 10^{-2}$	9.591
0.95	$-4.218\,829\,29 \times 10^{-1}$	$-4.621\,247\,00 \times 10^{-1}$	$4.024\,177\,03 \times 10^{-2}$	8.708

**Table 7.4.9.:** The total electron correlation for the infinite nuclear mass with selected non-integer charges.

#### **7.4.7. Effect of Correlation on Expectation Values**

Using the Laguerre polynomial-based wavefunction for the Hartree-Fock calculations, in addition to full three-body calculations, meant that expectation values could be calculated with existing techniques. Expectations values for the Hartree-Fock system were therefore calculated with 32 digits of precision using the HF wavefunction. From this the effect of correlation on the physical properties of the system in addition to the energies can be observed.

$H^-$	Expectation Value			
	Full 3Body	HF	3Body – HF	Electron Correlation Effect As % of 3Body
$\langle r_1 \rangle$	2.710 178 27	2.503 959 64	2.062 186 37 $\times 10^{-1}$	7.609
$\langle r_1^{-1} \rangle$	6.832 617 68 $\times 10^{-1}$	6.856 721 56 $\times 10^{-1}$	-2.410 388 00 $\times 10^{-3}$	-3.528 $\times 10^{-1}$
$\langle r_1^2 \rangle$	1.191 369 95 $\times 10^1$	9.411 097 45	2.502 602 03	2.101 $\times 10^1$
$\langle T \rangle$	5.277 510 17 $\times 10^{-1}$	2.439 648 67 $\times 10^{-1}$	2.837 861 49 $\times 10^{-1}$	5.377 $\times 10^1$
$\langle V \rangle$	-1.055 502 03	-4.879 297 34 $\times 10^{-1}$	-5.675 722 99 $\times 10^{-1}$	5.377 $\times 10^1$
$\langle \nu_{31} \rangle$	-9.999 999 70 $\times 10^{-1}$	-1.000 047 03	4.705 870 00 $\times 10^{-5}$	-4.706 $\times 10^{-3}$
$\langle \delta(r_1) \rangle$	1.645 528 74 $\times 10^{-1}$	1.796 349 93	-1.631 797 06	-9.917 $\times 10^2$
$\langle \eta \rangle$	-2.400 000 00 $\times 10^{-20}$	5.340 000 00 $\times 10^{-14}$		
$\langle \nu_{31} \rangle + Z_3$	2.964 870 00 $\times 10^{-8}$	-4.700 000 00 $\times 10^{-5}$		

**Table 7.4.10-a.:** Expectation values for the Hartree-Fock system  $H^-$  given in Table 7.4.8 compared to full three-body systems with explicit  $r_3$  interactions. The difference between these two gives the effects of the electron correlation on these physical properties.

He	Expectation Value			
	Full 3Body	HF	3Body – HF	Electron Correlation Effect As % of 3Body
$\langle r_1 \rangle$	$9.294\,722\,95 \times 10^{-1}$	$9.272\,734\,05 \times 10^{-1}$	$2.198\,890\,00 \times 10^{-3}$	$2.366 \times 10^{-1}$
$\langle r_1^{-1} \rangle$	1.688 316 80	1.687 282 22	$1.034\,585\,00 \times 10^{-3}$	$6.128 \times 10^{-2}$
$\langle r_1^2 \rangle$	1.193 483 00	1.184 828 48	$8.654\,515\,00 \times 10^{-3}$	$7.251 \times 10^{-1}$
$\langle T \rangle$	2.903 724 38	1.430 840 00	1.472 884 38	$5.072 \times 10^1$
$\langle V \rangle$	-5.807 448 75	-2.861 680 00	-2.945 768 76	$5.072 \times 10^1$
$\langle \nu_{31} \rangle$	-1.999 999 17	-1.999 999 43	$2.580\,210\,00 \times 10^{-7}$	$-1.290 \times 10^{-5}$
$\langle \delta(r_1) \rangle$	1.810 429 30	3.768 460 78	-1.958 031 49	$-1.082 \times 10^2$
$\langle \eta \rangle$	$1.800\,000\,00 \times 10^{-20}$	$4.530\,000\,00 \times 10^{-17}$		
$\langle \nu_{31} \rangle + Z_3$	$8.281\,870\,00 \times 10^{-7}$	5.700 000 00		

**Table 7.4.10-b.:** Expectation values for the Hartree-Fock system He given in Table 7.4.8 compared to full three-body systems with explicit  $r_3$  interactions. The difference between these two gives the effects of the electron correlation on these physical properties.

$C^{4+}$	Expectation Value				Electron Correlation Effect As % of 3Body
	Full 3Body	HF	3Body - HF		
$\langle r_1 \rangle$	$2.66793869 \times 10^{-1}$	$2.66757615 \times 10^{-1}$	$3.62534000 \times 10^{-5}$		$1.359 \times 10^{-2}$
$\langle r_1^{-1} \rangle$	5.68761533	5.68748294	$1.32381000 \times 10^{-4}$		$2.328 \times 10^{-3}$
$\langle r_1^2 \rangle$	$9.57395229 \times 10^{-2}$	$9.57019250 \times 10^{-2}$	$3.75980000 \times 10^{-5}$		$3.927 \times 10^{-2}$
$\langle T \rangle$	$3.24062466 \times 10^1$	$1.61805964 \times 10^1$	$1.62256502 \times 10^1$		$5.007 \times 10^1$
$\langle V \rangle$	$-6.48124932 \times 10^1$	$-3.23611929 \times 10^1$	$-3.24513003 \times 10^1$		$5.007 \times 10^1$
$\langle \nu_{31} \rangle$	-5.99999205	-5.99999962	$7.57204000 \times 10^{-6}$		$-1.262 \times 10^{-4}$
$\langle \delta(r_1) \rangle$	$6.14435736 \times 10^1$	$1.17554291 \times 10^1$	$4.96881445 \times 10^1$		$8.087 \times 10^1$
$\langle \eta \rangle$	$-3.50000000 \times 10^{-12}$	$1.90000000 \times 10^{-20}$			
$\langle \nu_{31} \rangle + Z_3$	$7.95000000 \times 10^{-6}$	$3.77000000 \times 10^{-7}$			

**Table 7.4.10-c.:** Expectation values for the Hartree-Fock system  $C^{4+}$  given in Table 7.4.8 compared to full three-body systems with explicit  $r_3$  interactions. The difference between these two gives the effects of the electron correlation on these physical properties.



Ar <sup>16+</sup>	Expectation Value				Electron Correlation Effect As % of 3Body
	Full 3Body	HF	3Body – HF		
$\langle r_1 \rangle$	$8.511\,040\,72 \times 10^{-2}$	$8.510\,926\,17 \times 10^{-2}$	$1.145\,480\,00 \times 10^{-6}$		$1.346 \times 10^{-3}$
$\langle r_1^{-1} \rangle$	$1.768\,751\,33 \times 10^1$	$1.768\,749\,83 \times 10^1$	$1.496\,700\,00 \times 10^{-5}$		$8.462 \times 10^{-5}$
$\langle r_1^2 \rangle$	$9.684\,480\,58 \times 10^{-3}$	$9.684\,119\,55 \times 10^{-3}$	$3.610\,350\,00 \times 10^{-7}$		$3.728 \times 10^{-3}$
$\langle T \rangle$	$3.129\,071\,86 \times 10^2$	$1.564\,305\,32 \times 10^2$	$1.564\,766\,54 \times 10^2$		$5.001 \times 10^1$
$\langle V \rangle$	$-6.258\,143\,72 \times 10^2$	$-3.128\,610\,63 \times 10^2$	$-3.129\,533\,09 \times 10^2$		$5.001 \times 10^1$
$\langle \nu_{31} \rangle$	$-1.799\,998\,09 \times 10^1$	$-1.799\,999\,98 \times 10^1$	$1.883\,810\,00 \times 10^{-5}$		$-1.047 \times 10^{-4}$
$\langle \delta(r_1) \rangle$	$1.788\,542\,81 \times 10^3$	$3.575\,174\,14 \times 10^1$	$1.752\,791\,07 \times 10^3$		$9.800 \times 10^1$
$\langle \eta \rangle$	$-1.230\,000\,00 \times 10^{-12}$	$-2.820\,000\,00 \times 10^{-22}$			
$\langle \nu_{31} \rangle + Z_3$	$1.906\,580\,00 \times 10^{-5}$	$2.280\,000\,00 \times 10^{-7}$			

**Table 7.4.10-d.:** Expectation values for the Hartree-Fock system Ar<sup>16+</sup> given in Table 7.4.8 compared to full three-body systems with explicit  $r_3$  interactions. The difference between these two gives the effects of the electron correlation on these physical properties.

Nuclear Charge of $Z_{\text{cr}} 0.911\,028$	Expectation Value			
	Full 3Body	HF	3Body - HF	Electron Correlation Effect As % of 3Body
$\langle r_1 \rangle$	4.146 663 48	2.989 003 77	1.157 659 72	$2.792 \times 10^1$
$\langle r_1^{-1} \rangle$	5.781 094 71 $\times 10^{-1}$	5.957 991 38 $\times 10^{-1}$	-1.768 966 70 $\times 10^{-2}$	-3.060
$\langle r_1^2 \rangle$	3.974 547 48 $\times 10^1$	1.397 688 13 $\times 10^1$	2.576 859 35 $\times 10^1$	$6.483 \times 10^1$
$\langle T \rangle$	4.149 862 23 $\times 10^{-1}$	1.869 531 21 $\times 10^{-1}$	2.280 331 02 $\times 10^{-1}$	$5.495 \times 10^1$
$\langle V \rangle$	-8.299 724 46 $\times 10^{-1}$	-3.739 062 41 $\times 10^{-1}$	-4.560 662 05 $\times 10^{-1}$	$5.495 \times 10^1$
$\langle \nu_{31} \rangle$	-9.109 876 78 $\times 10^{-1}$	-9.111 906 20 $\times 10^{-1}$	2.029 420 00 $\times 10^{-4}$	$-2.228 \times 10^{-2}$
$\langle \delta r_1 \rangle$	1.190 942 88 $\times 10^{-1}$	1.625 763 92	-1.506 669 63	$-1.265 \times 10^3$
$\langle \eta \rangle$	-5.900 000 00 $\times 10^{-21}$	2.200 000 00 $\times 10^{-14}$		
$\langle \nu_{31} \rangle + Z_3$	4.055 751 31 $\times 10^{-5}$	-1.623 842 71 $\times 10^{-4}$		

**Table 7.4.10-e.:** Expectation values for the Hartree-Fock system  $Z_{\text{cr}} (= 0.911\,028)$  given in Table 7.4.8 compared to full three-body systems with explicit  $r_3$  interactions. The difference between these two gives the effects of the electron correlation on these physical properties.

## 7. Electron Correlation in Two Electron Atoms

In the non-relativistic domain the effects electron correlation has on the expectation value depends on the value in question. However, in general, the effects of correlation upon the expectation values are least pronounced in high charge systems and these effects converge towards a unique value. As nuclear charge increases and the nucleus draws the two electrons closer, the influence of the correlation between the two electrons becomes smaller and smaller. At this point the nuclear attraction has become the dominant force pulling the electrons closer together and the electrons are no longer unable to move away from each other significantly.

In previous chapters the expectation values  $\langle \eta \rangle$  and  $\langle \nu_{31} \rangle$  have been a guide to the quality of a wavefunction (Section 4.4.3 on page 123). The value of  $\langle \eta \rangle$  indicates a high quality wavefunction when  $\langle \eta \rangle = 0$ . This is the virial condition and indicates the quality of the wavefunction over all regions of space. The expectation value  $\langle \nu_{31} \rangle$  is ideal when it equals  $Z_3$ , which is the cusp condition, indicating the quality of the wavefunction around the cusp, and is one of the harder properties to accurately describe as it is a divergent point in the wavefunction.<sup>[105]</sup> Both of these conditions can indicate that the wavefunctions is of good quality when fulfilled. Tables 7.4.10-a to 7.4.10-e gives these values and reveals that both the full three-body and the HF wavefunctions fulfil these conditions well.

As another indicator of quality, we consider the expectation values compared to literature. In Table 7.4.11 the HF expectation values are compared to literature (See Chapter 4 for discussion of the FC wavefunctions literature comparison). Not many accurate expectation values are available for the HF wavefunction, however the following Table 7.4.11 shows those available and are compared to those calculated here. The agreement with this work is very good indicating that the wavefunctions can accurately describe physical properties. Except in the case of  $H^-$  the energies calculated here are lower (more stable) than the literature values. As the HF method is variational, this means we have the best energies and therefore the best wavefunctions for calculating the expectation values.

System	Property	This work	Literature
$\text{H}^-$	$\langle r_1 \rangle$	2.503 959 636 938	2.5221 [222]
	$\langle r_1^{-1} \rangle$	0.685 672 155 949	0.6847 [222]
	$\langle r_1^2 \rangle$	9.411 097 448 798	9.6970 [222]
	$\langle \delta(r_1) \rangle$	0.154 597 341 126	0.154 64 [222]
	$\langle \hat{H} \rangle$	-0.487 929 734 369	-0.487 929 734 [214]
He	$\langle r_1 \rangle$	0.927 273 404 731	0.927 273 [223]
	$\langle r_1^{-1} \rangle$	1.687 282 215 281	1.6875 [224]
	$\langle r_1^2 \rangle$	1.184 828 479 909	1.1848 [222]
	$\langle \delta(r_1) \rangle$	1.797 959 103 761	1.797 980 73 [219]
	$\langle \hat{T} \rangle$	1.430 839 997 806	1.430 856 4 [225]
	$\langle \hat{V} \rangle$	-2.861 679 995 612	-2.861 696 35 [225]
	$\langle \hat{H} \rangle$	-2.861 679 995 612	-2.861 679 995 537 [219]
$\text{Li}^+$	$\langle r_1 \rangle$	0.572 366 815 001	0.5720 [222]
	$\langle r_1^{-1} \rangle$	2.687 419 466 644	2.6858 [222]
	$\langle r_1^2 \rangle$	0.445 320 566 922	0.445 18 [222]
	$\langle \delta(r_1) \rangle$	5.761 536 977 650	6.837 02 [222]
	$\langle \hat{T} \rangle$	3.618 207 600 726	3.618 109 5 [225]
	$\langle \hat{V} \rangle$	-7.236 415 201 452	-7.336 316 5 [225]
	$\langle \hat{H} \rangle$	-7.236 415 201 452	-7.236 415 2 [218]

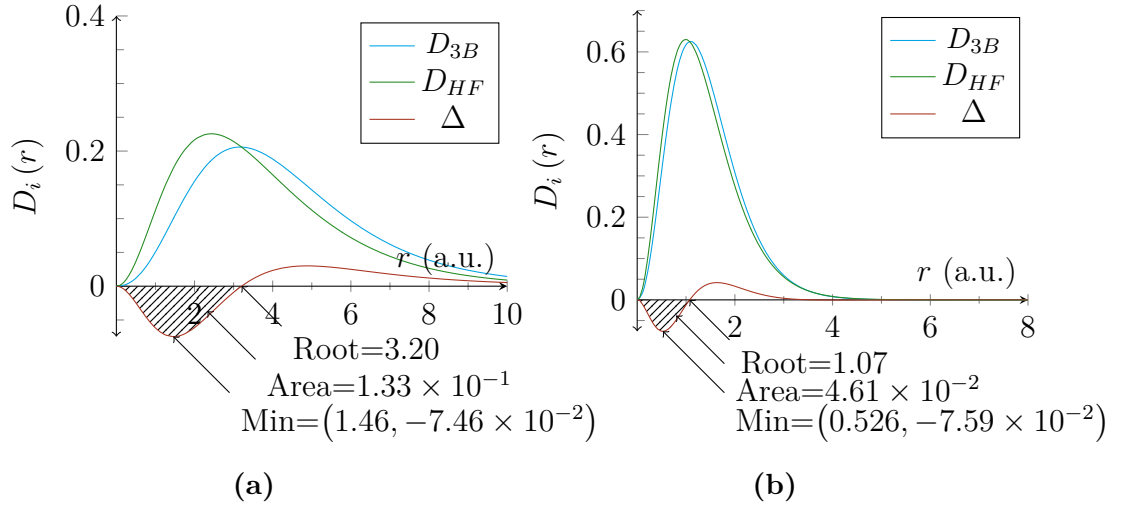
**Table 7.4.11.:** A comparison of calculated HF expectation values against available literature

### 7.4.8. Electron Densities and Electron Holes

The ground state electron density of a system can completely describe that system. This is the basis for density functional theory (DFT).<sup>[84,92,226–228]</sup> The difference between the electron density calculated using HF theory and that calculated using the FC method is a measure of the effects of electron correlation on the electron density. These effects include the coulomb hole,<sup>[184,229–232]</sup> which is caused by the instantaneous repulsive interaction between electrons. This decreases the probability of finding electrons near each other in the FC calculations but is not seen in the average repulsion model of the Hartree-Fock method. This effect has been known since before the 1960's. In recent literature,<sup>[187]</sup> as the accuracy of computations has increased, a second, smaller, coulomb hole may have been isolated by Gill *et al.*<sup>[187]</sup> This second coulomb hole requires very accurate particle densities in order to detect its presence. The wavefunctions used here should be accurate enough to also observe such an effect.

In this section the calculated radial distribution functions along the  $r_3$  coordinate are presented in Figure 7.4.4, for a few systems, using HF and FC methods. The difference between these two methods ( $\Delta$ ) represents the coulomb hole. The radial distribution functions are calculated using Eq. (7.4.1)

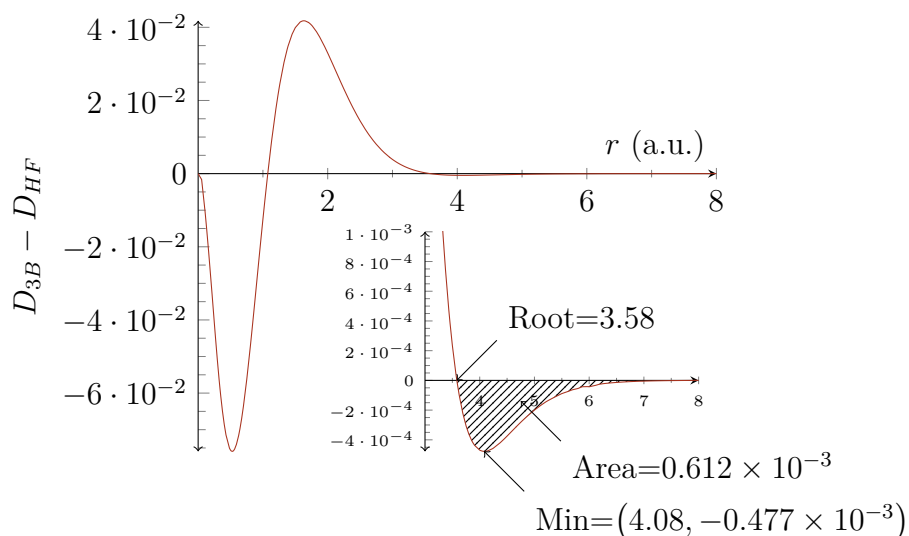
$$D(r) = 4\pi r^2 \langle \delta(r_3 - r) \rangle \quad (7.4.1)$$



**Figure 7.4.4.:** The electron densities of (a)  $\text{H}^-$  (b)  $\text{He}$  using both the FC method and the Hartree-Fock method. The difference between the two is also displayed and this shows the presence of the Coulomb hole.

The coulomb holes labelled in Figure 7.4.4 constitute the “no fly zone” for electrons. Here the coulomb repulsion between the two electrons keeps them apart from each other. Such a feature is only observable when comparing wavefunctions with and without electron correlation. The holes calculated here match those in the literature in terms of area, minima and root.<sup>[187,231]</sup> To investigate the presence of a second hole<sup>[187]</sup> we need an accurate HF and full three-body wavefunction. From Sections 7.4.2, 7.4.4 and 7.4.7 we know that we have accurate HF wavefunctions and Section 7.4.4 verified the accuracy of our three-body wavefunctions. Particularly of importance is the cusp condition as this indicates the quality of the wavefunction as a measure of particle density at the origin. Figure 7.4.5 shows that our wavefunction also reveals the presence of this second coulomb hole. This Figure 7.4.5 is an enlarged Figure 7.4.4b.

## 7. Electron Correlation in Two Electron Atoms



**Figure 7.4.5.:** The coulomb hole for He as given in Figure 7.4.4b but with the second coulomb hole brought into the forefront.

The second coulomb hole in Figure 7.4.5 matches that in the literature as shown in Table 7.4.12. This is important because, as noted in the original reference, this second coulomb hole may be a numerical artefact. The closely matching values from two separate methods suggests that this may be a physical phenomenon. To fully confirm this however we would require a convergence study of the hole itself to determine the rate of convergence of these values.

The area displaced by the coulomb hole corresponds to the charge displaced.<sup>[184]</sup> We can see that the second coulomb hole only moves a small amount of charge compared to the first, emphasising how small a contribution this hole makes to the effects of correlation on the electron density.

As a final note on this additional coulomb hole, Gill *et al.*<sup>[187]</sup> postulated that additional coulomb holes may exist. In the He wavefunction between the ranges  $0 a_0$  to  $20 a_0$  a total of 3 coulomb holes were identified, each with a maximum following it to give a total of 5 roots in the wavefunction. However these values cannot be confirmed without a convergence study as they are particularly tiny. It does suggest that the electron correlation effect has a repeating pattern of a minimum, followed by a root then a maximum. The characteristics of this are considered as interesting future work.

	Value	$Z_3$ -Scaled Value	$Z_3$ -Scaled Lit. Value <sup>[187]</sup>
Min	3.582	7.163	7.170
Root	4.086	8.171	8.170
area	$4.777 \times 10^{-4}$	$9.554 \times 10^{-4}$	$1.220 \times 10^{-3}$

**Table 7.4.12.:** The second He coulomb hole values compared to literature

### 7.4.9. Hartree-Fock Critical Nuclear Charge

In Chapter 6 it was shown, that when a second electron attempts to bind to a two-body atom, there is a minimum nuclear charge for which binding is possible.<sup>[28,126,147,176]</sup> We present here the critical charge of an atom (with infinite nuclear mass) for a Hartree-Fock wavefunction. This HF critical nuclear charge was calculated using a direct computation of the ionisation energy and changing  $Z_3$  until this ionisation energy is zero, using the built-in Maple routine `fsolve`. This routine solves an expression to zero, numerically, to the desired number of digits (in this case 31). Here it, numerically, solved the expression  $E_{\text{HF}} - E_{\text{th}} = 0$  in terms of  $Z_3$ . The critical nuclear charge and a variety of expectation values of such a system are given in Table 7.4.13. For an atom with two electrons under the clamped nucleus approximation the lowest continuum threshold is the two-body hydrogenic energy. This is known exactly as (for a ground state system  $n$  is 1):

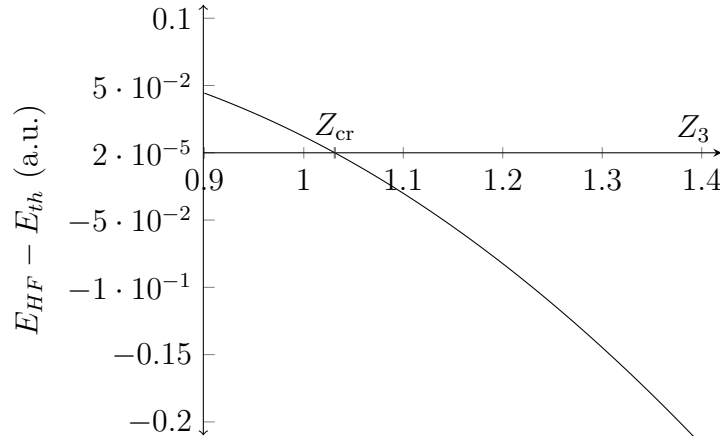
$$E_{th} = \frac{-Z^2}{2n} \quad (7.4.2)$$

For the fully-correlated system the critical nuclear charge is 0.911 028 (Chapter 6). We already know that the critical binding charge within HF is higher than this because the first continuum threshold for  $\text{H}^-$  according to Eq. (7.4.2) is  $-\frac{1}{2}$ , and the exact HF is  $-0.487\,929\,734$ .<sup>[218]</sup> From the exact HF energies and the two-body energy formulae it is evident that  $Z_{\text{cr}}$  is greater than 1 and lower than 2. This value differs from the critical nuclear charge given in Chapter 6 due to the absence of dynamic coulomb electron correlation effects in HF. The HF wavefunction underestimates the energy of a system by over estimating the effect of electron repulsion. As a result more nuclear charge is required to bind these electrons. According to Table 7.4.13 the critical charge is greater than unity and  $\text{H}^-$  is therefore unstable with respect to



Property	Value
$Z_3$	1.031 177 528
$\langle H_{HF} \rangle$	-0.531 663 547
$\langle H_H \rangle$	-0.531 663 547
$\langle r_1 \rangle$	2.372 691 816
$\langle r_1^2 \rangle$	8.370 837 14
$\langle r_1^{-1} \rangle$	0.717 056 38
$\langle T \rangle$	0.265 831 774
$\langle V \rangle$	-0.531 663 547
$\langle \nu_{31} \rangle$	-1.031 210 773
$\langle \delta(r_1) \rangle$	0.173 489 974

**Table 7.4.13.:** Properties of the critical nuclear charge system according to Hartree-Fock theory.  $\langle H_H \rangle$  is the hydrogenic energy, which corresponds to the threshold energy



**Figure 7.4.6.:** Graph of  $E_{HF} - E_{th}$  against  $Z$ . The crossing point on the x-axis is the critical nuclear charge for binding according to HF theory

the energy of the lowest continuum threshold energy. Since in reality  $H^-$  is stable, the importance of accurate electron correlation energy is made clear, even though, as we saw in Section 7.4.6, this correlation energy is only a small proportion of the total energy.

The HF  $Z_{cr}$  can be seen graphically as a function of  $E_{HF}(Z) - E_{th}(Z)$  as shown in Figure 7.4.6. The  $Z_{cr}$  is the crossing point on the x-axis of this graph.

## 7.5. Conclusion

To conclude, we have created an accurate HF wavefunction using Laguerre polynomials as our basis functions. This produced accurate HF wavefunctions with accurate

## 7. Electron Correlation in Two Electron Atoms

energies in agreement to the literature of 10 s.f. or more. This was combined with our own three-body wavefunction to calculate the effect dynamic electron correlation ( $E_{\text{corr}}$ ) has on our systems. For low-charged systems, this electron correlation constitutes a substantial fraction of the total energy, but this fraction decreases as charge increases. We also calculated the effect correlation has on expectation values where the inclusion of  $E_{\text{corr}}$  causes the electrons to move closer to the nucleus. For more complex properties such as the cusp, when the nucleus and electron coalesce it was found that low charges, as in He and below, the probability of coalescence was less likely in FC than in the HF method. At nuclear charges greater than two, the reverse was true with FC theory predicting greater chances of coalescence. From this we conclude that at low nuclear charge, the electrons in the FC method keep away from each other, but in the HF method they cannot since there is no dynamic correlation. At high nuclear charge the electrons in the FC method are found closer than in a HF calculation by using dynamic correlation to avoid each other but cannot in the HF method.

We characterised the coulomb hole of He and  $\text{H}^-$ , identifying aspects such as minima, root and area, where area corresponds to charge displaced by this hole. For the case of He we also confirmed the existence of the second coulomb hole and the possible existence of further holes beyond this in a repeating pattern of minimum, root, then maximum, and again root. The sequence then cycles back to minimum with decreasing magnitude as the cycle permutes. These other holes however require a deeper characterisation and accuracy study.

These results show us the effects dynamic electron correlation has on the electron density. In DFT this component is approximated, and this work like that of Colle and Salvetti,<sup>[190,191]</sup> with more work may be able to be used to formulate a new functional. Such a formulation would benefit from modern accurate data including non-integer charges that would help to give greater detail on electron correlation behaviour at low nuclear charge. This formulation is considered interesting future work.

Finally we presented the critical nuclear charge of the HF wavefunction as in the lowest charge where two electrons can bind to an infinitely massed nucleus. This charge is higher than the charge of  $\text{H}^-$  and exemplifies the importance of electron

## 7. *Electron Correlation in Two Electron Atoms*

correlation to predict real-world results.

## 8. Summary and Conclusions

The focus of this thesis has been the high accuracy computation of three-body systems without recourse to either the Born-Oppenheimer (BO) approximation or approximation of the like-charged particle interaction, which for the case of atoms corresponds to the electron correlation.

To pursue the best possible results and maximise performance, programs in various languages were developed uniquely for this work. These were designed to be efficient, extendable and, importantly, to calculate highly accurate energies, expectation values and wavefunctions. This was confirmed in Chapter 4.

The primary focus has been upon the stability of these three-body systems which has been facilitated by the full treatment of particles in motion, allowing for the effects of mass to be fully investigated. A complete stability boundary for any unit charge, three-body, system was presented using our accurate methodology. A possible emergence point from atomic to molecular structure was identified as the system with least fractional excess binding energy. This lies in the range identified by the centre-of-mass particle densities of Mátyus.<sup>[100]</sup>

The minimum nuclear charge required to bind two electrons to an atom, for various nuclear masses, was elucidated using a new variational method. This method is more efficient than any previous method in that it requires only one calculation to find this critical nuclear charge. An exact wavefunction would give the exact minimal charge, whereas any approximate wavefunction gives an upper bound to this charge. The nature of the electron as it becomes unbound was studied with a partitioning of the particle densities to separate out inner and outer electrons. This is the first time such partitioning has been performed using an accurate, fully correlated wavefunction and the first time it has been applied for the purpose of studying electron detachment at the critical charge. At the critical nuclear charge, it was shown that the inner electron

## 8. Summary and Conclusions

behaved almost exactly like an uncorrelated hydrogenic wavefunction but the outer electron still appears to be bound at a large radial distance of approximately  $6 a_0$ .

An accurate Hartree-Fock (HF) methodology was implemented as a reference function for the purpose of extracting accurate electron correlation ( $E_{\text{corr}}$ ). In most cases the HF energies were found to be at or near the HF limit as determined by available literature numerical HF data. The accurate model was developed such that non-integer low  $Z_3$  nuclear charges could be calculated to determine electron correlation for a wider range of systems than is typically calculable with the integer charge isoelectronic series. The effects of electron correlation on the energies and the expectation values such as  $\langle r_1 \rangle$ , the electron-nucleus average distance, was studied. For low charge systems, as in  $Z_3 < 2$ , where the electron-electron interaction dominates over the attractive nucleus-electron interaction, the amount of electron correlation energy relative to the total energy was high. At these low charges the electron density was more diffuse in the fully correlated system where the full inclusion of dynamic correlation meant that the electrons could better avoid each other. The electrons typically stay further away from the nucleus than in HF calculations which do not include dynamic correlation. For high charge systems, as in  $Z_3 > 2$ , where the nuclear-electron attraction dominates over the electron-electron repulsion, the relative electron correlation becomes smaller as the charge increases until it converges as  $Z \rightarrow \infty$ . With high nuclear charge systems, where it is more advantageous for electrons to be as close to the nucleus as possible the fully correlated system with dynamic electron correlation was able to demonstrate the electrons get closer to the nucleus than in the HF system in which this dynamic correlation is absent. Therefore dynamic correlation allows a system to respond by giving it the flexibility needed to minimise the energy by balancing becoming further from the other electron and closer to the nucleus as required by the nuclear charge.

The coulomb hole was also identified using our accurate methodology of a fully correlated wavefunction and an accurate HF reference wavefunction, where the difference between the two densities of these methods allowed us to characterise the coulomb hole accurately. Interestingly we also observed a secondary coulomb hole as recently first calculated by Gill, confirming its presence even with highly accurate methodologies. In accordance with Gill's prediction we also observed possible third

## 8. *Summary and Conclusions*

and fourth coulomb holes at the edge of detection accuracy.

The work here provides an understanding of the nature of three-body systems as the smallest unit capable of modelling both atomic and molecular systems. Electron correlation, the effects of electron detachment, stability as a function of mass and many other properties were elucidated for the purpose of being both interesting and providing accurate, usable results that may prove useful for the computational modeller and the chemist.

## A. Numbering Scheme

Three reduced index schemes are used in this work. They collapse the indices of the Laguerre polynomials in the wavefunction as  $l$ ,  $m$  and  $n$  to a single index. In its most general case it collapses all combinations of  $l$ ,  $m$  and  $n$  to a number. In the other two forms it takes account of wavefunction symmetry such that switching  $l$  and  $m$  does not change the number. The wavefunctions and matrices are ordered such that the  $i^{\text{th}}$  term or row/column corresponds to the  $l$ ,  $m$  and  $n$  given by the reduced index. These reduced indices are calculated as follows:

$$f(c, r) = \binom{c + r - 1}{r} \quad (\text{A.0.1a})$$

$$w = l + m + n \quad (\text{A.0.1b})$$

For the general asymmetric wavefunction case

$$k_{\text{asym}}(l, m, n) = f(l + m + n, 3) + f(l + m, 2) + f(l, 1) + 1 \quad (\text{A.0.1c})$$

For the symmetric wavefunction case

$$k_{\text{sym}}(l, m, n) = \frac{1}{24}w(w + 2)(2w + 5) + \frac{19}{16} - \frac{1}{16}(-1)^w + \frac{3}{2}l + \frac{1}{2}m + \frac{1}{4}(l + m)^2 - \frac{1}{8}(-1)^{l+m} \quad (\text{A.0.1d})$$

For the antisymmetric wavefunction case

$$k_{\text{antisym}}(l, m, n) = w(w + 2)(2 * w - 1) \frac{1}{24} - \frac{1}{16} + (-1)^w \frac{1}{16} + l(m + n) + m \quad (\text{A.0.1e})$$

## B. Energy Scaling and Charge Inversion

The coulomb Hamiltonian scales with various changes to the system in a predictable manner. For example scaling all particles masses by 10 scales the energy of the Hamiltonian by the same amount. Consider the typical coulomb Hamiltonian:

$$\hat{H} = \sum_i \frac{\nabla_i^2}{2m_i} + \sum_{i<j} \frac{q_i q_j}{r_{ij}} \quad (\text{B.0.1})$$

If all masses are multiplied by  $m$ , all charges by  $q$  and the distances scaled by  $r$  the kinetic energy gains the factor  $m^{-1}r^{-2}$  and the potential energy gains  $q^2r^{-1}$ . As a result the energy scales by the factor  $\epsilon = mq^4$ . Since charge scales the energy by  $q^4$  charge inversion will not change the energy and therefore a system with charges of  $Z_1 = 1, Z_2 = 1, Z_3 = -1$  and of  $Z_1 = -1, Z_2 = -1, Z_3 = 1$  will have the same energy. For a deeper discussion consult the review article by Armour *et al.*<sup>[28]</sup>



## C. Concavity of the Ground State

As noted by Armour *et al.*<sup>[28]</sup> and by Richard and Korobov<sup>[140]</sup> any property ( $\lambda$ ) that enters the Hamiltonian linearly is a concave function of the energy in the ground state. The second order derivative, as given by perturbation theory of the energy of a system, is always negative. Although this doesn't always hold, such as in radical excitations, it does hold for the ground state.<sup>[28]</sup>

## D. Arbitrary Nuclear Charge in Gaussian

This method in the Maple implementation using arbitrary non-integer charge is a trivial matter as any value of  $Z$  can be specified with no difficulties. However the Gaussian program makes several checks and assumptions which would normally forbid non-integer  $Z$ . In Gaussian 09 revision D.01 an additional keyword was added called `Znuc` which allowed for the specification of non-integer  $Z$ . It is worth noting however that the Gaussian program fails to correctly calculate the initial guess for systems with non-integer  $Z$ . In cases with non-integer  $Z$  it crashes during initial guess assignment with an error stating that the charge and number of electrons are in disagreement. This however can be circumvented by reading the initial guess directly from a previously calculated check point file. This check point file is generated from a calculation in which the `Znuc` keyword is not used. The following is an example com file of a `Znuc` calculation:

The Gaussian com file for arbitrary nuclear charge of  $\text{H}^-$

```
1 %oldchk=H--.chk
2 %chk=H-_ZNuc.chk
3 #p hf/aug-cc-pV6Z guess=read scf=VeryTightLinEq Integral=(Acc2E=8)
4
5 IsoElectronic -1 ZNuc=0.9
6
7 -1 1
8 H(ZNuc=0.9)                0.00000000    0.00000000    0.00000000
9
10
```

## E. HF 2 Electron Integral Symmetry

This combined index takes  $u, v, p$  and  $q$  and returns a unique index for each unique integral. This is done by considering the indices to be the indexes of a 4-dimensional symmetric matrix where we extract only one half of this symmetric matrix.

$$r = \frac{u(u+1)}{2} + v \quad (\text{E.0.1})$$

$$s = \frac{p(p+1)}{2} + q \quad (\text{E.0.2})$$

$$\text{UniqueIndex}(u, v, p, q) = \frac{r(r+1)}{2} + s \quad (\text{E.0.3})$$

The number of two electron integrals to calculate is reduced to  $\frac{n(n+1)(n^2+n+2)}{8}$  by this technique. In this step these rules are used to calculate all two electron integrals and store them for later use.

## F. Licences

### F.1. Grand Central Dispatch - Apache License

Copyright (c) 2008–2013 Apple Inc. All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License");  
you may not use this file except in compliance with the License.  
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software  
distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or  
implied.

See the License for the specific language governing permissions and  
limitations under the License.

### F.2. MPFRC++ - General Public License

MPFR C++ (MPREAL): Multiple precision floating point arithmetic library for  
C++.

Thread-safe, cross-platform (MSVC, GCC, ICC), one-header C++ library.  
Supports C++11 features if available, C++03 compatible otherwise.

Thin wrapper for MPFR: <http://mpfr.org>

Project homepage: <http://www.holoborodko.com/pavel/mpfr>

Contact e-mail: [pavel@holoborodko.com](mailto:pavel@holoborodko.com)

Copyright (c) 2008–2015 Pavel Holoborodko

Contributors:

Dmitriy Gubanov, Konstantin Holoborodko, Brian Gladman,  
Helmut Jarausch, Fokko Beekhof, Ulrich Mutze, Heinz van Saanen,  
Pere Constans, Peter van Hoof, Gael Guennebaud, Tsai Chia Cheng,  
Alexei Zubanov, Jauhien Piatlicki, Victor Berger, John Westwood,  
Petr Aleksandrov, Orion Poplawski, Charles Karney, Arash Partow,  
Rodney James, Jorge Leitao.

Licensing:

(A) MPFR C++ is under GNU General Public License ("GPL").

(B) Non-free licenses may also be purchased from the author, for users who do not want their programs protected by the GPL.

The non-free licenses are for users that wish to use MPFR C++ in their products but are unwilling to release their software under the GPL (which would require them to release source code and allow free redistribution).

Such users can purchase an unlimited-use license from the author.

Contact us for more details.

GNU General Public License ("GPL") copyright permissions statement:

\*\*\*\*\*

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or

(at your option) any later version.

This program is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU General Public License for more details.

You should have received a copy of the GNU General Public License  
along with this program. If not, see <<http://www.gnu.org/licenses/>>.

### **F.3. MPFR - lesser General Public License**

Copyright 1999–2015 Free Software Foundation, Inc.  
Contributed by the AriC and Caramel projects, INRIA.

This file is part of the GNU MPFR Library.

The GNU MPFR Library is free software; you can redistribute it and/or modify  
it under the terms of the GNU Lesser General Public License as published by  
the Free Software Foundation; either version 3 of the License, or (at your  
option) any later version.

The GNU MPFR Library is distributed in the hope that it will be useful, but  
WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY  
or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General  
Public  
License for more details.

You should have received a copy of the GNU Lesser General Public License  
along with the GNU MPFR Library; see the file COPYING.LESSER. If not, see  
<http://www.gnu.org/licenses/> or write to the Free Software Foundation, Inc.,  
51 Franklin St, Fifth Floor, Boston, MA 02110–1301, USA. \*/

## **F.4. Doublepresso - MIT License**

Copyright (c) 2008–2009, Petr Kobalicek <kobalicek.petr@gmail.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **F.5. ASMJit - Zlib License**

AsmJit — Complete x86/x64 JIT and Remote Assembler for C++  
Copyright (c) 2008–2015, Petr Kobalíček kobalicek.petr@gmail.com

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

This notice may not be removed or altered from any source distribution.

## **F.6. Numerical Recipes Personal Single-User License**

The following license terms and the Disclaimer of Warranty, below, apply to individual purchases of the Numerical Recipes Code download product, and to purchases of the Numerical Recipes Code CD-ROM.

By purchasing this disk or code download, you acquire a Numerical Recipes Personal Single-User License. This license lets you personally use Numerical Recipes code ("the code") on any number of computers, but only one computer at a time. You are not permitted to allow anyone else to access or use the code. You may, under this license, transfer precompiled, executable applications incorporating the code to other, unlicensed, persons, providing that (i) the application is noncommercial (e.g., does not involve the selling or licensing of the application for a fee or its use in developing commercial products or services), and



(ii) the application was first developed, compiled, and successfully run by you, and (iii) the code is bound into the application in such a manner that it cannot be accessed as individual routines and cannot practicably be unbound and used in other programs. That is, under this license, your application user must not be able to use Numerical Recipes code as part of a program library or "mix and match" workbench.

Businesses and organizations that purchase this disk or code download, and that thus acquire one or more Numerical Recipes Personal Single–User Licenses, may permanently assign those licenses, in the number acquired, to individual employees. Such an assignment must be made before the code is first used and, once made, it is irrevocable and can not be transferred. In many cases, business and organizations may instead wish to purchase Numerical Recipes institutional subscriptions (see below), which have more flexibility.

## **F.7. Lapack - Modified BSD License**

Copyright (c) 1992–2013 The University of Tennessee and The University of Tennessee Research Foundation. All rights reserved.

Copyright (c) 2000–2013 The University of California Berkeley. All rights reserved.

Copyright (c) 2006–2013 The University of Colorado Denver. All rights reserved.

Additional copyrights may follow

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.
- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

The copyright holders provide no reassurances that the source code provided does not infringe any patent, copyright, or any other intellectual property rights of third parties. The copyright holders disclaim any liability to any recipient for claims brought against recipient by any third party for infringement of that parties intellectual property rights.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT

LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR

A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT

OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,

SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT

LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,

DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED

AND ON ANY  
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,  
OR TORT  
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT  
OF THE USE  
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH  
DAMAGE.

## **F.8. Boost Software License**

Boost Software License — Version 1.0 — August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the "Software") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third—parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine—executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY  
KIND, EXPRESS OR  
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY,  
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON—  
INFRINGEMENT. IN NO EVENT  
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE

SOFTWARE BE LIABLE  
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT,  
TORT OR OTHERWISE,  
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE  
OR THE USE OR OTHER  
DEALINGS IN THE SOFTWARE.

## **F.9. MacOpt - lesser General Public License**

Copyright (c) 2002 David J.C. MacKay and Steve Waterhouse and Mark Gibbs

This library is free software; you can redistribute it and/or  
modify it under the terms of the GNU Lesser General Public  
License as published by the Free Software Foundation; either  
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
GNU

Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public  
License along with this library; if not, write to the Free Software  
Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111–1307 USA

GNU licenses are here :

<http://www.gnu.org/licenses/licenses.html>

Author contact details are here :

<http://www.inference.phy.cam.ac.uk/mackay/c/macopt.html> mackay@mrao.cam.  
ac.uk

## **F.10. GAlib - MIT License**

Copyright 1995–1996 Massachusetts Institute of Technology (MIT)

all rights reserved

Copyright 1996–1999 Matthew Wall (the Author)

all rights reserved

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

You may copy and distribute the source code and/or library/executable code for GAlib in any medium provided that you conspicuously and appropriately give credit to the author and keep intact all copyright and disclaimer notices in the library.

Any publications of work based upon experiments that use GAlib must include a suitable acknowledgement of GAlib. A suggested acknowledgement is: "The software for this work used the GAlib genetic algorithm package, written by Matthew Wall at the Massachusetts Institute of Technology."

The author of GAlib and MIT assume absolutely no responsibility for the use or misuse of GAlib. In no event shall the author of GAlib or MIT be liable for any damages resulting from use or performance of GAlib.

# Bibliography

- [1] I. N. Levine, *Physical Chemistry*, McGraw-Hill Education, New York, 4th edn., 2009.
- [2] M. Pahlavani and S. Motevalli, *Acta Physica Polonica B*, 2009, **40**, 319–329.
- [3] W. Breunlich, *Annual Review of Nuclear and Particle Science*, 1989, **39**, 311–356.
- [4] K. Ishida, K. Nagamine, T. Matsuzaki and N. Kawamura, *Journal of Physics G-Nuclear and Particle Physics*, 2003, **29**, 2043–2045.
- [5] V. R. Bom, A. M. Demin, D. L. Demin, C. W. E. van Eijk, M. P. Faifman, V. V. Filchenkov, A. N. Golubkov, N. N. Grafov, S. K. Grishechkin, K. I. Gritsaj, V. G. Klevtsov, A. D. Konin, A. V. Kuryakin, S. V. Medved, R. K. Musyaev, V. V. Perevozchikov, A. I. Rudenko, S. M. Sadetsky, Y. I. Vinogradov, A. A. Yukhimchuk, S. A. Yukhimchuk, V. G. Zinov and S. V. Zlatoustovskii, *Journal of Experimental and Theoretical Physics*, 2005, **100**, 663–687.
- [6] D. V. Balin, V. A. Ganzha, S. M. Kozlov, E. M. Maev, G. E. Petrov, M. A. Soroka, G. N. Schapkin, G. G. Semenchuk, V. A. Trofimov, A. A. Vasiliev, A. A. Vorobyov, N. I. Voropaev, C. Petitjean, B. Gartner, B. Lauss, J. Marton, J. Zmeskal, T. Case, K. M. Crowe, P. Kammel, F. J. Hartmann and M. P. Faifman, *Physics of Particles and Nuclei*, 2011, **42**, 185–214.
- [7] A. M. Frolov, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2001, **34**, 3813–3828.
- [8] A. M. Frolov, *European Physical Journal D*, 2012, **66**, 1–11.
- [9] L. I. M. Somov and L. N., *Soviet Physics Uspekhi*, 1990, **33**, 616–646.
- [10] A. M. Frolov and D. M. Wardlaw, *European Physical Journal D*, 2011, **63**, 339–350.
- [11] A. Deltuva, A. C. Fonseca and P. U. Sauer, *Annual Review of Nuclear and Particle Science*, 2008, **58**, 27–49.
- [12] J. Randazzo, F. Buezas, A. Frapiccini, F. Colavecchia and G. Gasaneo, *Physical Review A*, 2011, **84**, 052715.
- [13] C. McCurdy, M. Baertschy and T. Rescigno, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2004, **37**, R137–R187.
- [14] E. Alt, B. Irgaziev and A. Mukhamedzhanov, *Modern Physics Letters A*, 2005, **20**, 947–963.

## Bibliography

- [15] Y. Kim and A. Zubarev, *Physical Review A*, 1997, **56**, 521–526.
- [16] E. Alt and A. Mukhamedzhanov, *Physical Review A*, 1993, **47**, 2004–2022.
- [17] H. Bethe and E. Salpeter, *Quantum mechanics of one and two electron atoms*, Dover Publications, 1957.
- [18] P. J. Mohr, B. N. Taylor and D. B. Newell, *Reviews of Modern Physics*, 2012, **84**, 1527–1605.
- [19] K. Konishi and G. Paffuti, *Quantum mechanics*, Oxford University Press, New York, 2009.
- [20] E. Schrödinger, *Physical Review*, 1926, **28**, 1049–1070.
- [21] E. Schrödinger, *Collected Papers on Wave Mechanics*, Blackie & Son Limited, London, 1928.
- [22] R. K. Prasad, *Quantum Chemistry*, New Age International, 2006.
- [23] J. Simons, *An Introduction to Theoretical Chemistry*, Cambridge University Press, 2003.
- [24] M. Born and R. Oppenheimer, *Annalen der Physik*, 1927, **389**, 457–484.
- [25] B. T. Sutcliffe and R. G. Woolley, *The Journal of Chemical Physics*, 2012, **137**, 22A544.
- [26] L. Pauling and E. B. Wilson, *Introduction to Quantum Mechanics*, McGraw Hill Book Company Inc., London, 1935.
- [27] E. Lewars, *Computational Chemistry*, Springer Science & Business Media, 2010.
- [28] E. A. G. Armour, J. Richard and K. Varga, *Physics Reports*, 2005, **413**, 1–90.
- [29] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C : the art of scientific computing*, Cambridge University Press, Cambridge, 1st edn., 1988.
- [30] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical recipes : the art of scientific computing*, Oxford University Press, Cambridge, 3rd edn., 2007.
- [31] B. Schaling, *The Boost C++ Libraries*, XML Press, Laguna Woods, 2nd edn., 2014.
- [32] Apple Inc., *libdispatch.macosforge.org*, 2009, Apache License.
- [33] M. J. D. Powell, *The BOBYQA algorithm for bound constrained optimization without derivatives*, Cambridge NA Report 2009/NA06, University of Cambridge, 2009.
- [34] A. R. Conn, K. Scheinberg and L. N. Vicente, *Introduction to Derivative-free Optimization*, SIAM, 2009.

## Bibliography

- [35] P. Holoborodko, *holoborodko.com/pavel/mpfr/*, 2008, GNU General Public License.
- [36] G. Hanrot, V. Lefèvre, P. Pélicier, P. Théveny and P. Zimmermann, *ACM Trans Math Software*, 2007, **33**, 13–13.
- [37] D. H. Bailey, Y. Hida, X. S. Li, B. Thompson, K. Jeyabalan and A. Kaiser, *crd-legacyblgov/ dhbailey/mpdist/*, 2000, modified BSD license.
- [38] A. Tartakynov, *github.com/tartakynov/mathpresso*, 2012, MIT license.
- [39] E. W. Dijkstra, *Annual Review in Automatic Programming*, 1963, **3**, 329–345.
- [40] G. Ballard, J. Demmel, O. Holtz and O. Schwartz, *SIAM Journal on Scientific Computing*, 2010, **32**, 3495–3523.
- [41] I. Bar-On, *Parallel Computing*, 1991, **17**, 409–417.
- [42] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edn., 1999.
- [43] M. Nakata, *mplapack.sourceforge.net/*, 2010, BSD License.
- [44] D. E. King, *Journal of Machine Learning Research*, 2009, **10**, 1755–1758.
- [45] D. F. Shanno and Kettler, P. C., *Mathematics of Computation*, 1970, **24**, 657–664.
- [46] D. F. Shanno, *Mathematics of Computation*, 1970, **24**, 647–656.
- [47] R. H. Byrd, P. Lu, J. Nocedal and C. Zhu, *SIAM Journal on Scientific Computing*, 1995, **16**, 1190–1208.
- [48] D. MacKay, *inference.phy.cam.ac.uk/mackay/c/macopt.html*, 2002, GNU Lesser General Pulic License.
- [49] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.
- [50] P. Becker, *The C++ Standard Library Extensions*, Pearson Education, 2006.
- [51] Y. Hardy, K. S. Tan and W. H. Steeb, *Computer Algebra with SymbolicC++*, World Scientific, 2008.
- [52] F. P 'erez and B. E. Granger, *Computing in Science and Engineering*, 2007, **9**, 21–29.
- [53] SymPy Development Team, *sympy.org*, 2007, New BSD license.
- [54] P. E. Regier and A. J. Thakkar, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1984, **17**, 3391–3403.



## Bibliography

- [55] A. J. Thakkar and V. H. Smith, *Journal of Chemical Physics*, 1977, **67**, 1191–1196.
- [56] O. Tange, *login: The USENIX Magazine*, 2011, **36**, 42–47.
- [57] R. T. Pack and W. B. Brown, *The Journal of Chemical Physics*, 1966, **45**, 556–559.
- [58] A. M. Frolov, *Chemical Physics Letters*, 2015, **626**, 49–54.
- [59] D. P. Chong and D. M. Schrader, *Molecular Physics*, 1969, **16**, 137–144.
- [60] E. Mátyus, J. Hutter, U. Müller-Herold and M. Reiher, *The Journal of Chemical Physics*, 2011, **135**, 204302.
- [61] P. A. M. Dirac, *The principles of quantum mechanics*, 1947.
- [62] C. Schwartz, *International Journal of Modern Physics E*, 2006, **15**, 877–888.
- [63] E. A. Hylleraas, *Zeitschrift for Physik*, 1929, **54**, 347–366.
- [64] H. Hettema, *Quantum Chemistry*, World Scientific, 2014.
- [65] H. Nakashima and H. Nakatsuji, *The Journal of Chemical Physics*, 2008, **128**, 154107.
- [66] C. E. Moore, *Ionization Potentials and Ionization Limits Derived from the Analyses of Optical Spectra*, United States. Government Printing Office, Washington D.C., 1970.
- [67] J. Bartlett, J. Gibbons and C. Dunn, *Physical Review*, 1935, **47**, 679–680.
- [68] J. Bartlett, *Physical Review*, 1937, **51**, 0661–0669.
- [69] C. Pekeris, *Physical Review*, 1962, **127**, 509–511.
- [70] C. Pekeris, *Physical Review*, 1959, **115**, 1216–1221.
- [71] C. Pekeris, *Physical Review*, 1958, **112**, 1649–1658.
- [72] A. A. Frost, *The Journal of Chemical Physics*, 1964, **41**, 478–482.
- [73] H. Cox, S. Smith and B. Sutcliffe, *Physical Review A*, 1994, **49**, 4520–4532.
- [74] F. Arias de Saavedra, E. Buendia, F. J. Gálvez and A. Sarsa, *European Physical Journal D*, 1998, **2**, 181–190.
- [75] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Courier Corporation, 2012.
- [76] W. Boyce and R. DiPrima, *Elementary Differential Equations and Boundary Value Problems*, Wiley, New York, 7th edn., 2001.
- [77] A. Jeffrey and H. H. Dai, *Handbook of Mathematical Formulas and Integrals*, Academic Press, 2008.

## Bibliography

- [78] E. W. Weisstein, *Laguerre Polynomial*. From *MathWorld*, A Wolfram Web Resource, 2015.
- [79] I. C. Society, *IEEE Standard for Floating-Point Arithmetic*, IEEE Std 754-2008 ISSNVO, Washington, D.C., 2008.
- [80] E. W. Weisstein and D. Lichtblau, *Condition Number*. From *MathWorld*, A Wolfram Web Resource, 2014.
- [81] E. W. Weisstein and W. K. Macura, *Numerical Stability*. From *MathWorld*, A Wolfram Web Resource, 2014.
- [82] A. Ermolaev and G. Sochilin, *Doklady Akademii Nauk Sssr*, 1964, **155**, 1050.
- [83] K. Frankowski and C. Pekeris, *Physical Review*, 1966, **146**, 46–49.
- [84] B. Bransden and C. Joachain, *Physics of Atoms and Molecules*, Addison Wesley, 1983.
- [85] A. C. King, J. Billingham and S. R. Otto, *Differential Equations*, Cambridge University Press, 2003.
- [86] H. Cox, P. Sinclair, S. J. Smith and B. Sutcliffe, *Molecular Physics*, 1996, **87**, 399–406.
- [87] H. Cox, S. Smith and B. Sutcliffe, *Physical Review A*, 1994, **49**, 4533–4539.
- [88] H. Cox, *Ph.D. thesis*, University of York, 1994.
- [89] G. Drake, *Nuclear Instruments & Methods in Physics Research Section B-Beam Interactions with Materials and Atoms*, 1988, **31**, 7–13.
- [90] I. S. Dhillon and B. N. Parlett, *SIAM Journal on Matrix Analysis & Applications*, 2003, **25**, 858–899.
- [91] I. S. Dhillon and B. N. Parlett, *Linear Algebra and its Applications*, 2004, **387**, 1–28.
- [92] P. Atkins and J. De Paula, *Atkins' Physical Chemistry*, Oxford University Press, Oxford, 8th edn., 2006.
- [93] F. Arias de Saavedra, E. Buendia, F. J. Gálvez and A. Sarsa, *European Physical Journal D*, 2001, **13**, 201–206.
- [94] P. J. Mohr, B. N. Taylor and D. B. Newell, *arXiv.org*, 2008, 633–730.
- [95] D. H. Bailey and A. M. Frolov, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2002, **35**, 4287–4298.
- [96] R. E. Moss, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1999, **32**, L89.
- [97] A. Martin, J.-M. Richard and T. T. Wu, *Physical Review A*, 1992, **46**, 3697–3703.

## Bibliography

- [98] A. M. Frolov and V. Smith, *Physical Review A*, 1994, **49**, 3580–3585.
- [99] S. Bhattacharyya, J. K. Saha, P. K. Mukherjee and T. K. Mukherjee, *Physica Scripta*, 2012, **85**, 065305.
- [100] E. Mátyus, J. Hutter, U. Müller-Herold and M. Reiher, *Physical Review A*, 2011, **83**, 52512.
- [101] D. R. Hartree, Proceedings of the Cambridge Philosophical Society, 1928, p. 426.
- [102] V. Korobov, *Physical Review A*, 2000, **61**, 064503.
- [103] G. B. Arfken and H. J. Weber, *Mathematical Methods for Physicists*, Academic Press, 2013.
- [104] W. Thirring, *A Course in Mathematical Physics*, Springer-Verlag, 1981, vol. 3.
- [105] T. Kato, *Communications on Pure and Applied Mathematics*, 1957, **10**, 151–177.
- [106] E. Steiner, *The Journal of Chemical Physics*, 1963, **39**, 2365–2366.
- [107] A. M. Frolov, *Physical Review A*, 1999, **59**, 4270–4280.
- [108] F. Desaaavedra, I. Porras, E. Buendia and F. J. Gálvez, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1995, **28**, 3123–3133.
- [109] A. M. Frolov, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2002, **35**, L331.
- [110] A. M. Frolov and V. H. Smith, *The Journal of Chemical Physics*, 2003, **119**, 3130–3137.
- [111] A. M. Frolov, *Journal of Chemical Physics*, 2007, **126**, 104302–104302.
- [112] A. M. Frolov, *Physical Review A*, 2004, **69**, 022505.
- [113] G. Drake, M. Cassar and R. Nistor, *Physical Review A*, 2002, **65**, 054501.
- [114] J.-P. Karr and L. Hilico, *Phys. Rev. Lett.*, 2012, **109**, 103401.
- [115] M. Rotenberg and J. Stein, *Physical Review*, 1969, **182**, 1–7.
- [116] E. A. G. Armour and D. M. Schrader, *Canadian Journal of Physics*, 1982, **60**, 581–583.
- [117] R. D. Poshusta, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1985, **18**, 1887.
- [118] A. M. Frolov and D. M. Bishop, *Physical Review A*, 1992, **45**, 6236–6240.
- [119] T. K. Rebane and A. V. Filinsky, *Physics of Atomic Nuclei*, 2002, **65**, 41–48.
- [120] T. K. Rebane and A. A. Kuzminskii, *Optics and Spectroscopy*, 2002, **93**, 833–836.

- [121] D. K. Gridnev and C. Greiner, *Phys. Rev. Lett.*, 2005, **94**, 223402.
- [122] D. K. Gridnev, C. Greiner and W. Greiner, *Journal of Mathematical Physics*, 2005, **46**, 052104.
- [123] F. H. Gertler, H. B. Snodgrass and L. Spruch, *Physical Review*, 1968, **172**, 110–118.
- [124] I. Aronson, C. J. Kleinman and L. Spruch, *Physical Review A*, 1971, **4**, 841–846.
- [125] A. W. King, F. Longford and H. Cox, *The Journal of Chemical Physics*, 2013, **139**, 224306.
- [126] T. K. Rebane, *Soviet Physics Journal of Experimental and Theoretical Physics*, 1990, **71**, 1055–1064.
- [127] T. K. Rebane, *Optics and Spectroscopy*, 2003, **95**, 888–894.
- [128] T. K. Rebane, *Physics of Atomic Nuclei*, 2009, **72**, 55–58.
- [129] T. K. Rebane, *Physics of Atomic Nuclei*, 2009, **72**, 1465–1474.
- [130] A. E. Yavuz, Ö. Masalci and N. Kazanci, *NANOTR9, 9th Nanoscience and Nanotechnology Conference Special Issue*, 2014, **318**, 251–255.
- [131] M. A. Rahman, C. Ho, W. Gierlotka and J. Kuo, *Journal of Electronic Materials*, 2014, **43**, 4582–4593.
- [132] M. Premović, D. Manasijević, D. Minić and D. Živković, *Materials Chemistry and Physics*, 2014, **148**, 356–363.
- [133] J. J. Ruan, C. P. Wang, S. Y. Yang, R. Kainuma and X. J. Liu, *Journal of Alloys and Compounds*, 2014, **615**, 998–1005.
- [134] T. Omori, K. Makino, K. Shinagawa, I. Ohnuma, R. Kainuma and K. Ishida, *Intermetallics*, 2014, **55**, 154–161.
- [135] E. Özduğan, B. Andak, N. Bulutçu, S. Türdü and M. Tatlier, *Fluid Phase Equilibria*, 2014, **381**, 67–70.
- [136] A. W. King, P. E. Herlihy and H. Cox, *Journal of Chemical Physics*, 2014, **141**, 044120.
- [137] E. A. G. Armour and W. B. Brown, *Acc. Chem. Res.*, 1993, **26**, 168–173.
- [138] R. N. Hill, *Journal of Mathematical Physics*, 1976, **18**, 2316–2330.
- [139] R. N. Hill, *Physical Review Letters*, 1977, **38**, 643–646.
- [140] V. Korobov and J. M. Richard, *Physical Review A*, 2005, **71**, 024502.
- [141] T. Chihara, *Proceedings of the American Mathematical Society*, 1957, **8**, 765–767.
- [142] S. Kar and P. Mandal, *Physical Review A*, 2000, **62**, 052514.

- [143] A. S. Kadyrov, I. Bray and A. T. Stelbovics, *Journal of Physics: Conference Series*, 2007, **88**, 012062.
- [144] S. Nayek and A. Ghoshal, *Physics of Plasmas (1994-present)*, 2012, **19**, 113501.
- [145] K. Higgins and P. G. Burke, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1991, **24**, L343.
- [146] R. N. Hewitt, C. J. Noble and B. H. Bransden, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1991, **24**, L635.
- [147] C. S. Estienne, M. Busuttil, A. Moini and G. W. F. Drake, *Physical Review Letters*, 2014, **112**, 173001.
- [148] J. D. Baker, D. E. Freund, R. N. Hill and J. D. Morgan, *Physical Review A*, 1990, **41**, 1247–1273.
- [149] N. L. Guevara and A. V. Turbiner, *Physical Review A*, 2011, **84**, 064501.
- [150] J. Zamastil, J. Cizek and M. Simanek, *Physical Review A*, 2010, **81**, 032118.
- [151] E. Hylleraas, *Z. Phys.*, 1928, **48**, 469–494.
- [152] T. Koga, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2014, **47**, 215001.
- [153] T. Koga, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2012, **45**, 185003.
- [154] H. Matsuyama and T. Koga, *Journal of Computational Applied Mathematics*, 2010, **233**, 1584–1589.
- [155] T. Koga and H. Matsuyama, *Theoretical Chemistry Accounts: Theory, Computation and Modeling*, 2007, **118**, 931–935.
- [156] T. Koga, *Theoretical Chemistry Accounts: Theory, Computation and Modeling*, 2007, **117**, 575–578.
- [157] H. Matsuyama and T. Koga, *Theoretical Chemistry Accounts: Theory, Computation and Modeling*, 2007, **118**, 643–647.
- [158] T. Koga and H. Matsuyama, *Theoretical Chemistry Accounts: Theory, Computation and Modeling*, 2006, **115**, 59–64.
- [159] T. Koga, *The Journal of Chemical Physics*, 2004, **121**, 3939–3940.
- [160] C. F. Fischer, *Computer Physics Communications*, 1991, **64**, 431–454.
- [161] C. F. Fischer, *Computer Physics Communications*, 1978, **14**, 145–153.
- [162] T. Koga and H. Matsuyama, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2011, **44**, 195003.
- [163] P. A. M. Dirac, *The Principles of Quantum Mechanics*, Oxford University Press, Oxford, 1981.

## Bibliography

- [164] M. J. O. Strutt, *Zeitschrift for Physik*, 1928, **48**, 162–164.
- [165] M. R. Spiegel, *Journal of Applied Physics*, 1952, **23**, 906–909.
- [166] H. L. Hamburger, *Mathematical Proceedings of the Cambridge Philosophical Society*, 1949, **45**, 489–494.
- [167] R. A. Johnson, *Proceedings of the Institute of Radio Engineers*, 1956, **44**, 1058–1059.
- [168] P. A. CLAVIER, *Proceedings of the Institute of Radio Engineers*, 1956, **44**, 1876–1877.
- [169] M. L. Boas, *Mathematical Methods in the Physical Sciences*, John Wiley & Sons, 2006.
- [170] M. Renardy and R. C. Rogers, *An Introduction to Partial Differential Equations*, Springer, 2004.
- [171] W. Kaplan, *Advanced Calculus*, Academic Internet Publications Inc., 3rd edn., 2007.
- [172] E. W. G. R. Weisstein, *XOR. From MathWorld*, A Wolfram Web Resource, 2015.
- [173] M. Wang, G. Audi, A. H. Wapstra, F. G. Kondev, M. MacCormick, X. Xu and B. Pfeiffer, *Chinese Physics C*, 2012, **36**, 1603.
- [174] H. Nakashima and H. Nakatsuji, *The Journal of Chemical Physics*, 2007, **127**, 224104.
- [175] A. M. Frolov, *Physical Review A*, 1998, **57**, 2436–2439.
- [176] A. W. King, L. C. Rhodes, C. A. Readman and H. Cox, *Physical Review A*, 2015, **91**, 042512.
- [177] Krijn, M. P. C. M. and D. Feil, *Chemical Physics Letters*, 1988, **150**, 45–54.
- [178] J. A. Pople, M. Head-Gordon and K. Raghavachari, *The Journal of Chemical Physics*, 1987, **87**, 5968–5975.
- [179] J. Tao, J. Perdew, V. Staroverov and G. Scuseria, *Physical Review Letters*, 2003, **91**, 146401–146404.
- [180] V. Fock, *Izvestiya Akademii Nauk, Seriya Fizicheskaya*, 1954, **18**, 161.
- [181] D. J. Tozer, N. C. Handy and W. H. Green, *Chemical Physics Letters*, 1997, **273**, 183–194.
- [182] P.-O. Löwdin, *Physical Review*, 1955, **97**, 1509–1520.
- [183] P.-O. Löwdin, *Physical Review*, 1955, **97**, 1474–1489.
- [184] C. A. Coulson and A. H. Neilson, *Proceedings of the Physical Society*, 1961, **78**, 831.

## Bibliography

- [185] V. W. Maslen, *Proceedings of the Physical Society. Section A*, 1956, **69**, 734–740.
- [186] P. M. W. Gill, D. P. O’Neill and N. A. Besley, *Theoretical Chemistry Accounts: Theory, Computation and Modeling*, 2003, **109**, 241–250.
- [187] J. K. Pearson, P. M. W. Gill, J. M. Ugalde and R. J. Boyd, *Molecular Physics*, 2009, **107**, 1089–1093.
- [188] P.-F. Loos and P. M. W. Gill, *Journal of Chemical Physics*, 2010, **132**, 234111.
- [189] J. W. Hollett, L. K. McKemmish and P. M. W. Gill, *The Journal of Chemical Physics*, 2011, **134**, 224103.
- [190] R. Colle and O. Salvetti, *Theoretica Chimica Acta*, 1975, **37**, 329–334.
- [191] C. Lee, W. Yang and R. G. Parr, *Physical Review B*, 1988, **37**, 785–789.
- [192] Y. Imamura and G. E. Scuseria, *The Journal of Chemical Physics*, 2003, **118**, 2464–2469.
- [193] J. C. Sancho-García and F. Moscardó, *The Journal of Chemical Physics*, 2003, **118**, 1054–1058.
- [194] Y. Watanabe and H. Tatewaki, *The Journal of Chemical Physics*, 2005, **123**, 074322.
- [195] S. Fraga and J. M. G. de la Vega, *Journal of Physics B: Atomic Molecular and Optical Physics*, 2005, **38**, 279.
- [196] G. Pestka, H. Tatewaki and J. Karwowski, *Physical Review A*, 2004, **70**, 024501.
- [197] D. Frye, A. Preiskorn and E. Clementi, *Journal of Computational Chemistry*, 1991, **12**, 560–564.
- [198] G. Drake, *Canadian Journal of Physics*, 1988, **66**, 586–611.
- [199] E. R. Davidson, S. A. Hagstrom, S. J. Chakravorty, V. M. Umar and C. F. Fischer, *Physical Review A*, 1991, **44**, 7071–7083.
- [200] H. Tatewaki and T. Noro, *Chemical Physics Letters*, 2004, **399**, 480–483.
- [201] H. D. R., *Mathematical Proceedings of the Cambridge Philosophical Society*, 1928, **24**, 111–132.
- [202] D. R. Hartree, *Mathematical Proceedings of the Cambridge Philosophical Society*, 2008, **24**, 89–110.
- [203] D. R. Hartree and W. Hartree, *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences*, 1935, **150**, 9–33.

- [204] M. Frisch, G. Trucks, H. Schlegel, G. Scuseria, M. Robb, J. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. Hratchian, A. Izmaylov, J. Bloino, G. Zheng, J. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. Montgomery, J. Peralta, F. Ogliaro, M. Bearpark, J. Heyd, E. Brothers, K. Kudin, V. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. Burant, S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. Millam, M. Klene, J. Knox, J. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. Stratmann, O. Yazyev, A. Austin, R. Cammi, C. Pomelli, J. Ochterski, R. Martin, K. Morokuma, V. Zakrzewski, G. Voth, P. Salvador, J. Dannenberg, S. Dapprich, A. Daniels, Farkas, J. Foresman, J. Ortiz, J. Cioslowski and D. Fox, *Gaussian 09, Revision D.01*, 2009.
- [205] D. Redfern, *The Maple handbook*, Springer, 1994.
- [206] D. Feller, *Journal of Computational Chemistry*, 1996, **17**, 1571–1586.
- [207] K. L. Schuchardt, B. T. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li and T. L. Windus, *Journal of Chemical Information and Modeling*, 2007, **47**, 1045–1052.
- [208] K. Pachucki and J. Komasa, *Chemical Physics Letters*, 2004, **389**, 209–211.
- [209] D. E. Woon and T. H. J. Dunning, *The Journal of Chemical Physics*, 1994, **100**, 2975–2988.
- [210] I. Gradshteyn, I. Ryzhik and A. Jeffrey, *Table of Integrals, Series, and Products*, Academic Press, San Diego, 6th edn., 2000.
- [211] T. G. e. al, *gmpilib.org*, 2002, LGPL.
- [212] C. F. Fischer, *Canadian Journal of Physics*, 1963, **41**, 1895–1910.
- [213] C. F. Fischer, *Canadian Journal of Physics*, 1968, **46**, 2336–2337.
- [214] C. J. C. Roothaan and G. A. Soukup, *International Journal of Quantum Chemistry*, 1979, **15**, 449–462.
- [215] K. Szalewicz and H. J. Monkhorst, *The Journal of Chemical Physics*, 1981, **75**, 5785–5788.
- [216] G. B. Bacskay, *Chemical Physics*, 1981, **61**, 385–404.
- [217] T. Koga, S. Watanabe, K. Kanayama, R. Yasuda and A. J. Thakkar, *The Journal of Chemical Physics*, 1995, **103**, 3000–3005.
- [218] T. Koga, H. Tatewaki and A. J. Thakkar, *The Journal of Chemical Physics*, 1994, **100**, 8140–8144.
- [219] de Saavedra, E. Buendia and F. A. F J Galvez, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1994, **27**, 5131–5137.



## Bibliography

- [220] T. Koga, M. Omura, H. Teruya and A. J. Thakkar, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1995, **28**, 3113.
- [221] Y. A. Bernard, P.-F. Loos and P. M. W. Gill, *Molecular Physics*, 2013, **111**, 2414–2426.
- [222] K. E. Banyard and C. C. Baker, *The Journal of Chemical Physics*, 1969, **51**, 2680–2689.
- [223] H. Tatewaki, T. Koga, Y. Sakai and A. J. Thakkar, *The Journal of Chemical Physics*, 1994, **101**, 4945–4948.
- [224] I. Porras and F. J. Gálvez, *Physical Review A*, 1992, **46**, 105–112.
- [225] E. Clementi and C. Roetti, *Atomic Data and Nuclear Data Tables*, 1974, **14**, 177–478.
- [226] P. Hohenberg and W. Kohn, *Physical Review*, 1964, **136**, B864–B871.
- [227] M. Levy, *Proceedings of the National Academy of Sciences*, 1979, **76**, 6062–6065.
- [228] S. F. Sousa, P. A. Fernandes and M. J. Ramos, *The Journal of Physical Chemistry A*, 2007, **111**, 10439–10452.
- [229] M. P. Barnett, F. W. Birss and C. A. Coulson, *Molecular Physics*, 1958, **1**, 44–47.
- [230] N. Moiseyev, J. Katriel and R. Boyd, *Theoretica Chimica Acta*, 1977, **45**, 61–67.
- [231] R. F. Curl and C. A. Coulson, *Proceedings of the Physical Society*, 1965, **85**, 647.
- [232] R. J. Boyd and C. A. Coulson, *Journal of Physics B: Atomic Molecular and Optical Physics*, 1973, **6**, 782.